NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

OPTIMAL DIGITAL DETECTION OF ACOUSTIC SIGNALS IN COLORED NOISE

by

Martin A. Cloutier

December, 1995

Thesis Co-Advisors: Ralph Hippenstiel
Roberto Cristi

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1995	Master's Thesis

4. TITLE AND SUBTITLE

OPTIMAL DIGITAL DETECTION OF ACOUSTIC SIGNALS IN COLORED NOISE (U)

5. FUNDING NUMBERS

6. AUTHOR(S)

Cloutier, Martin André

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/ MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/ MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This thesis addresses optimal methods for the detection of acoustic signals corrupted by colored noise. In achieving this we provide a study of the characteristics of ambient noise in the ocean and the digital techniques which can be used in the process of detecting known acoustic signals which are corrupted by that noise. Various techniques are studied, in particular the use of matrix decomposition techniques applied to the correlation matrix or to a data matrix, and the matched filter for colored noise. Other methods such as the inverse filter, the differential operator, and the adaptive prediction-error filter will also be looked at for their whitening properties. The theoretical foundations of those techniques are presented as well as the application of each method to the problem. Simulations are conducted for each technique in order to provide quantified performance measurements supporting the use of each method.

14. SUBJECT TERMS

Detection, Ambient noise, Matrix decomposition, Colored noise.

15. NUMBER OF PAGES

159

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Approved for public release; distribution is unlimited.

OPTIMAL DIGITAL DETECTION OF ACOUSTIC SIGNALS IN COLORED NOISE

Martin André Cloutier
Captain, Canadian Armed Forces
B.Eng., Royal Military College of Canada, 1986

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
and
MASTER OF SCIENCE IN ENGINEERING ACOUSTICS

from the

NAVAL POSTGRADUATE SCHOOL
December 1995

Author: Martin André Cloutier

Approved by:

Ralph Hippenstiel, Thesis Co-Advisor

Roberto Cristi, Thesis Co-Advisor

Herschel H. Loomis, Jr., Chairman,
Department of Electrical and Computer Engineering

Anthony A. Atchley, Chairman,
Engineering Acoustics Academic Committee

ABSTRACT

This thesis addresses optimal methods for the detection of acoustic signals corrupted by colored noise. In achieving this we provide a study of the characteristics of ambient noise in the ocean and the digital techniques which can be used in the process of detecting known acoustic signals which are corrupted by that noise. Various techniques are studied, in particular the use of matrix decomposition techniques applied to the correlation matrix or to a data matrix, and the matched filter for colored noise. Other methods such as the inverse filter, the differential operator, and the adaptive prediction-error filter will also be looked at for their whitening properties. The theoretical foundations of those techniques are presented as well as the application of each method to the problem. Simulations are conducted for each technique in order to provide quantified performance measurements supporting the use of each method.

TABLE OF CONTENTS

ACKNOWLEDGMENT

I wish to acknowledge my two thesis advisors, Professor Ralph Hippenstiel and Professor Roberto Cristi, for their invaluable help and insight in helping me in the research and writing of this thesis. Their frequent discussions of the research and thought-provoking ideas on unresolved issues often paved the way to the next challenge. Furthermore, their class notes provided me with a firm basis on which to rest the foundation of this thesis.

I could not have completed this thesis research without the support of my wife Lucie who spent countless hours entertaining and taking care of my son Maxime and daughter Myriam while I enjoyed the relative peace of school and conducted my studies. Maxime and Myriam also are to be thanked for the enjoyment they have given me while at home. Had I not had these diversions, my thesis work would have been far less pleasurable. I thank my family most sincerely for being there for me.

I. INTRODUCTION

A. THESIS GOAL

This thesis is concerned with the detection of periodic signals as generally emitted by the machinery of travelling vessels whether submerged or at the surface of the ocean. The goal is to provide a study of the discrete time techniques that can be used for the optimal detection of acoustic signals in colored noise. To achieve this goal, a comprehensive study of the ambient noise present in the ocean is discussed as well as the effect on this noise of the recording, sampling and digitization of the input data. Having identified the characteristics of the ambient noise presented to the detector algorithm, this thesis then studies various transformation techniques to process the correlated samples received in order to achieve an optimal detection of the desired signal.

B. ENVIRONMENT

The environment in which underwater signals are generated, propagated and intercepted is of great importance to the techniques of signal processing used in analyzing propagating sounds. We are concerned with the underwater properties of the world's oceans. The environment of our oceans in terms of ambient noise, is characterized by the propagating medium, namely sea water. Water in itself profoundly changes the spectral characteristics of any broadband sound propagating in it, in that it absorbs acoustic energy in a way that is highly dependent on the frequency content of those signals. Sea water is particular in that it contains significant amounts of dissolved salts. Some of these salts also have important absorptive qualities that affect signals of various frequencies differently. These properties have a significant effect on the ambient noise to which all signals of interest are subjected to. This noise is quite different from the common assumptions of white noise. That is, the spectral shape of the ambient noise in the ocean is frequency dependent.

C. DISCRETE TIME DETECTION

The detection aspect of this work is accomplished through the use of probability and statistical tools to design a receiver which provides the user the ability to discriminate between signal plus noise and noise only. Through the use of modern technology, these signals are most often sampled in order to be processed with the digital processors in use today. Therefore, this thesis concentrates on discrete time detection, that is, detection using only discrete time quantities as input and output data. The theory of detection in continuous time is only addressed in those areas where it is useful in order to provide the reader with a greater understanding of discrete time detection. The receivers to be designed are optimum in that they best satisfy a given criteria such as signal to noise ratio, under a given set of assumptions.

D. TECHNIQUES

Because of the spectral shape of the ambient noise corrupting the signals, the techniques that are investigated are concerned with whitening the noise. This may be done in different ways. The matched filter for colored noise inherently whitens the noise samples and achieves detection in one step. Other optimal techniques require the use of a prewhitener and a white noise detector. Since we know the power spectral density (spectrum) of the noise, it is possible to create a filter which acts as an equalizer. Running the received signal through it has the effect of whitening the additive noise allowing the use of an optimal white noise detection scheme. This process is known as inverse filtering. Other techniques use a decomposed correlation matrix or covariance matrix to transform the input data vector from one consisting of both correlated signal and noise samples, to one with uncorrelated noise samples plus the signal. Various techniques have different advantages in terms of performance, complexity and computational effectiveness. The use of a data matrix to compute the orthogonal matrices is also addressed by using the Singular Value Decomposition (SVD) and QR methods.

E. THESIS OUTLINE

This thesis is divided into two main parts. The first part, addressed in Chapter II, provides a study of the processes that affect the detection of signals in the underwater acoustical environments. Acoustic signals in the ocean are necessarily corrupted by the ambient noise of the environment from which they have been recorded and in which they have been emitted. Furthermore, the means in which they are recorded further modifies the inherent characteristics of the noise corrupting the signal. In order to provide an optimal detection scheme for these signals, we must have a good understanding of the noise process, and the transmission of the noise and of the acoustic signals within the ocean.

The second major part is dealt with in Chapter III. This chapter provides the reader with basic detection theory as applied to the discrete case with colored noise as is prevalent in the ocean. The major obstacle to be overcome for the detection of signals in colored noise is that the samples of the noise are correlated. In order for most discrete time detection scheme to function, these samples must undergo a transformation which decorrelates the noise. The various procedures in which this decorrelation can be done will be documented. We will study the matched filter for colored noise and prewhitening of the noise using various factorization techniques. The correlation matrix, the covariance matrix or a data matrix will be factored into products of other matrices which have the required characteristics leading to uncorrelated noise samples. We will also address the use of the differential operator to whiten the noise, and, the whitening properties of the prediction-error filter.

Chapter IV presents the results of simulations undertaken with each of the techniques presented in Chapter III to provide comparative performance results. In each case, white noise discrete time detection of periodic signals is done through the use of the Fast-Fourier Transform (FFT) to compute averaged periodograms. First, the averaged periodogram is shown for the original sequence with the colored noise. Then the averaged periodogram of the whitened sequence is shown for the particular transformation under examination. This

will allow for an objective view of the result and permit us to make qualified statements on the performance of each technique.

II. ACOUSTIC AMBIENT NOISE SPECTRUM

The spectral characteristics of the noise affecting the detection performance of modern detectors and receivers are shaped by a variety of mechanisms. Figure 2.1 shows a block diagram of the various causes and transformations which have a part in shaping the noise corrupting received signals in acoustic environments.

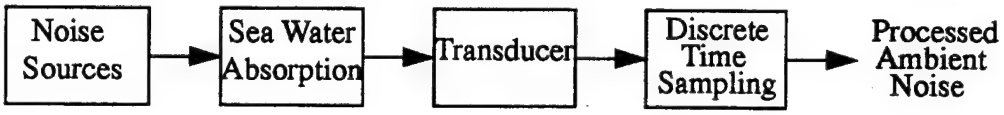

Figure 2.1: Generation and processing of ocean acoustic noise

A. NOISE SOURCES

The study of the ambient noise frequency spectrum begins with a consideration of the individual sources of noise. These sources are divided into the following general groups: Thermal agitation, hydrodynamic, man-made, seismic, biological, and polar. The effect of these sources cannot be completely studied without considering the effects of attenuation. Attenuation is defined as the loss of acoustic energy from a sound beam. It is divided in two parts: absorption mechanisms which convert acoustics energy into thermal energy as a result of the compressive/decompressive effects of the sound wave, and scattering effects which deflect energy out of the direction of propagation of the acoustic wave [Ref. 1].

1. Thermal Agitation Noise

The minimum noise level of a medium is determined by the effects of thermal agitation. For the ocean, in the temperature range of 0-30° C, the equivalent thermal-noise sound pressure level (SPL) in dB re $2 \times 10^{3} \mu Pa$ for a 1Hz bandwidth is given by [Ref. 2]

$$SPL_{TH} \approx -101 + 20\log f . \qquad (2.1)$$

The thermal noise spectral density derived from this equation is shown in Figure 2.2.

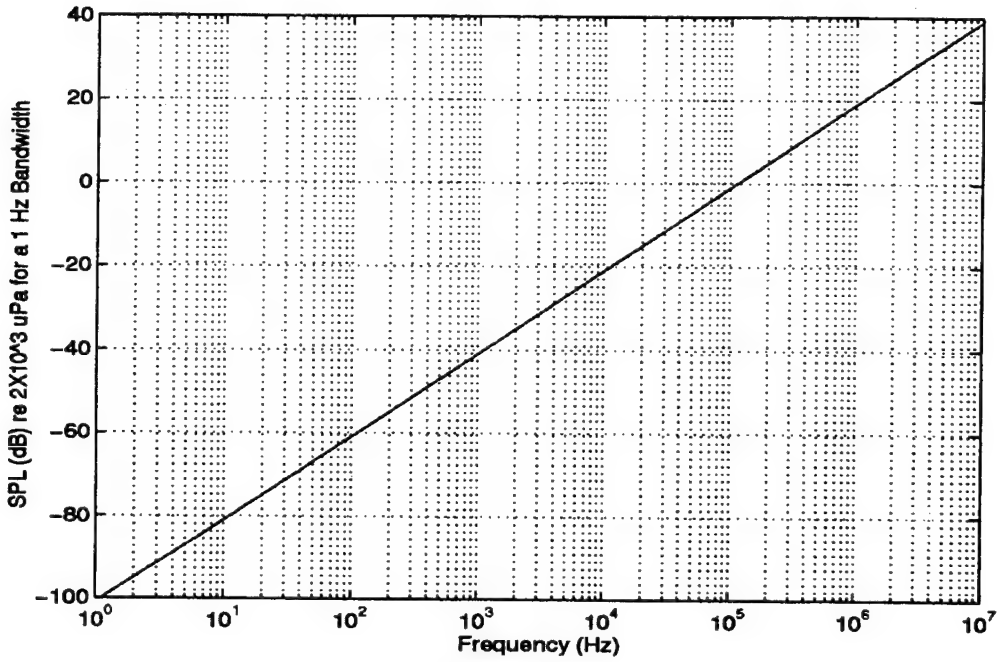

Figure 2.2: Thermal agitation sound pressure level (SPL)

According to this equation, the thermal noise spectrum has a level of -10 dB at 35 kHz and a positive slope of +6 dB/octave. At higher frequencies, 20-30 kHz and up, the observed minimum ambient noise levels are very close to the thermal noise levels shown on Figure 2.2. At lower frequencies ($f < 10$ kHz) however, a significant difference exists where the observed ambient noise levels are significantly higher than those which would result only from thermal noise.

2. Hydrodynamic Noise

This group contains the noise sources which are related to hydrodynamic processes. Many of these are important radiators of sound and contribute to the noise spectra in the ocean.

a. Air Bubbles

Air bubbles affect the ambient noise spectrum in two distinct ways. Their effect on the attenuation properties of the medium are discussed in a later paragraph. This

paragraph addresses the manner in which sound energy is radiated when air bubbles exist, and the cavitation noise resulting from their collapse. When wind effects are considered, substantial noise levels [Ref. 3] have been observed when air is entrained into the water by water droplets on the surface. However, bubbles do not only occur as a result of wind. Bubbles are also created as a result of biological effects such as decaying matter, sea floor gas seepage, fish and marine mammals. As air bubbles in the ocean are transported or simply rise to the surface, they are subjected to temperature and pressure fluctuations due to depth, turbulence, currents and waves. These variations induce oscillations in the bubbles causing them to emit noise at frequencies related to their diameters. It has been concluded [Ref. 3] that air bubbles and their cavitation produced at or near the surface as a result of the wind, are an important source of wind dependent noise in the frequency range of 50-10,000 Hz. The spectrum shape of cavitation noise is similar to that of the air bubble noise, and is characterized by a slope of approximately -6 dB/octave in the frequency range identified above. As the frequency is reduced, the noise level resulting from this effect drops-off with a 12 dB/octave slope due the lack of bubbles of a size large enough to resonate at those frequencies. Experimentation has shown that at shallow depth, both the noise spectrum level and its shape in frequencies between 50-10,000 Hz are likely to be mostly the result of resonating noise and cavitation noise from the air bubbles [Ref. 3].

b. Surface Waves

The performance of an underwater transducer is affected by the subsurface pressure fluctuations caused by changes in surface elevations. In the frequency range of 500 Hz to 20 kHz, the agitation of the sea surface is the primary source of ambient noise [Ref. 1]. Relations between sea state, mean wave height, and representative wind speed have been derived and are usually given in table formats such as that given at [Ref. 1: p. 414]. As a result, it is possible to characterize the ambient noise caused by surface waves, by specifying wind speed. In the range of frequencies specified above, it has been established that the noise spectrum level decreases at a rate of 17 dB/decade [Ref. 1].

c. Turbulence

Turbulence is defined by the instability of the water motion. It is usually created at boundary regions where water and solid surfaces interact or when fluid layers flow past one another with different velocities and directions. Turbulence between the transducer or the vessels containing it is generally considered to be self-noise. Turbulent effects usually consist of large volume fluctuations resulting in pressure variations at frequencies less than 1 Hz. Although some noise is created at higher frequencies when large-scale eddies breakup into smaller scale phenomena, it has been observed [Ref. 3] that these noise levels are several orders of magnitude below the ambient noise level observed. It is considered that the radiated noise caused by turbulence is not a contributing factor to the ambient noise level in the ocean. However, the pressure fluctuations do affect pressure transducers when located in a region of turbulence. Substantial ambient oceanic turbulence has been observed [Ref. 3], with pressure level spectrum occurring in the frequency range below 10 Hz. In swift oceanic streams, the pressure level can be affected up to 100 Hz, while extreme tidal currents can generate pressures at even higher frequencies. It has been concluded that the turbulent pressure fluctuations are an important component of noise below 10 Hz, and in some cases below 100 Hz, but that its contribution to the ambient noise spectrum from radiated noise is negligible [Ref. 3].

3. Man-made Sources

Ships are important sources of underwater sounds. At any given time, it is estimated that there are over 1000 ships under way in the North Atlantic Ocean [Ref. 4]. Rotational and reciprocating machinery is required for propulsion, control, and habitability. These components generate vibrations which are transmitted through the hull of the ship to the surrounding water. The propeller of the ships is a major source of this underwater noise. At lower frequencies (10 Hz to 250 Hz) than those associated with surface waves, a major source of noise is generated by distant shipping. The shape of the noise spectra and its strength resulting from this oceanic traffic will be affected by a combination of the

transmission loss, the number of ships, and the distribution of those ships. For example, a transducer in deep water in an open ocean may receive significant shipping noise from widely scattered ships because the average transmission loss will be relatively small. The ambient noise may be likewise considerably affected in a situation where transmission losses are high but ship concentration is large, in a shallow water location such as that occurring along coastal shipping lanes. Traffic noise may also depend on the types of ships involved, each of which may generate different noises. However, we assume that the number of ships affecting the typical acoustic location will blend the individual differences of the distant ships into an average source characteristics with Gaussian distribution.

For most surface ships, the effective source of the radiated noise is the engine/ propeller combination where the noise is generated between three and ten meters below the surface. This places the source in the shallow-water channel. As a result of the surface reflection, the source and its image will operate as an acoustic doublets radiating noise with a spectrum slope of +6 dB/octave relative to the spectrum of the simple source.

Studies of noise from surface ships have determined [Ref. 3] that the sound pressure level spectrum has a slope of about -6 dB/octave. However, the spectrum is highly variable below 1 kHz, and some tests have shown a general levelling of the spectra at the 100 Hz frequency.

Industrial activities may also locally affect the ambient noise characteristics. Many mechanical movements related to such things as oil drilling, pile driving, mining, etc., will transmit sounds to the water when located in an area close to shore or at sea such as on drilling platforms. These noises can be accounted for in the design of fixed transducers in their vicinity but are generally ignored in the design of other transducers due to their highly localized nature. Operators however must be aware of these sources.

4. Seismic Noise

The earth's crust is constantly moving as a result of volcanic and tectonic action. This action produces waves which are transmitted through the solid crust and eventually to the

water. Due to the nature of the crust and of the water, even far from the point of origin of a disturbance, appreciable amounts of energy can find their way to the ocean waters and be propagated as compressional waves. The same effect can results from man-made explosions. The spectral characteristics of such an event will depend on the magnitude of the disturbance, the range, and the propagation path. This noise is typically characterized by one or a series of transients of relative short duration. In general, the affected frequencies will be less than 500 Hz with a maximum between 2 and 20 Hz [Ref. 3]. Noticeable underwater sound will be generated from 1 to 100 Hz [Ref. 3]. Outside of the specific effect of the transients themselves, a background of continuous seismic noise is also present. This effect is attributed to the after effects of the main transients and to the effects of storms, and waves and swells hitting the shores along coastal zones. It was found that seismic noise dominates the ambient noise spectrum at very low frequencies between 1 and 100 Hz [Ref. 3]. However this noise spectrum level is of a transitory nature, and is highly dependent on time and location. Due to its nature, seismic noise is particularly important to the design and operation of bottom-mounted transducers.

5. Biological Sources

Underwater sounds originating from marine life can be an important source of noise at most frequencies of interest [Ref. 3]. The wide variety of sounds are usually of short duration but are often repeated. Continuous noise is often observed when it is the result of a great number of individuals such as shrimps. As with most other sources, the ambient noise of biological origin varies with location, time and species involved. In some case, the sounds are correlated with geographical, seasonal and diurnal pattern enabling users of systems to improve the performance of their systems by considering the effect of these sources. However, because of this variance, it is generally ignored in establishing the ambient noise spectrum of an area of ocean.

6. Polar Noise

Polar regions cause noise spectra which are very different than those prevalent in other oceans because of the effect of the ice covering large portions of the water. Cracking of the ice as a result of thermal effects, and movements of large pieces generate cracking, straining, crunching, sliding and percussion sounds. Other sounds are created by icebergs as they melt. As the ice changes into water, numerous pressurized air bubbles which were trapped in the ice, burst through the thinning surface. As the air rushes out, a sharp cracking sound is produced. The noise produced by these pressurized air bubbles has been found to be the prevalent source of noise when icebergs are present. These sounds are highly characteristic and must be addressed differently than more typical underwater noises due to their impulsive nature. Some methods dealing with this environment have been devised and generally include a provision for removing the impulsive components of the ambient noise prior to the whitening and detection process [Ref. 5]. However, the level and character of the noise spectra vary according to ice condition, wind speed, ice pack snow cover, and air temperature changes [Ref. 4]. Due to the extreme variations observed, it is not possible to predict the spectra and amplitude distribution of the noise to be expected in polar regions.

7. Estimated Noise Spectra

The nominal shape of the noise spectrum level for the deep-water case is shown in Figure 2.3.

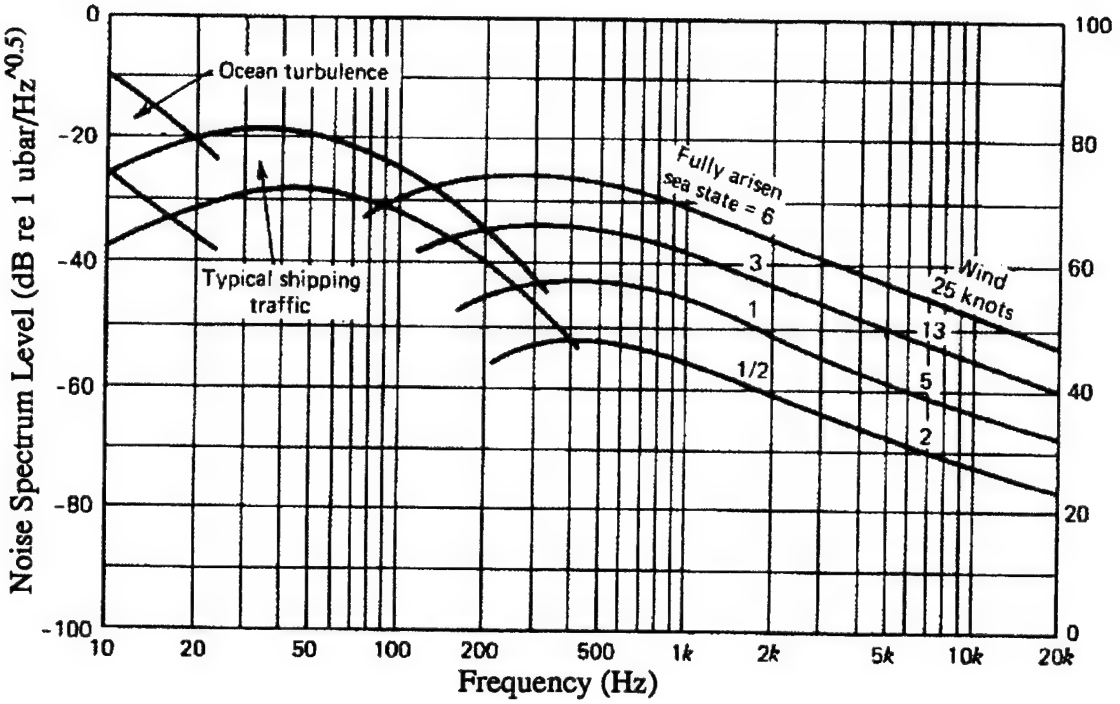

Figure 2.3: Deep-water ambient noise spectrum level. From Ref. [1].

Below 20 Hz, it has been found that ocean turbulence and seismic noise predominate. Above 20 Hz and until about 300 Hz, shipping noise and biological noises are the main contributor. From 300 Hz and above, surface noise highly dependent on wind speed provides most of the noise energy. Although not shown on this figure, at 50 kHz, thermal agitation of the water molecules overwhelms all other sources of noise and continues to increase with frequency at a rate of 6 dB/octave as shown in Figure 2.2.

8. Distribution of the Noise

It has been shown that the ambient noise spectra in the oceans is the result of a very large number of noise sources of various characteristics whose sounds propagate in and are affected by the underwater medium. Although some impulsive sources of underwater noises can be non-Gaussian and others may even have asymmetric distributions, by the central limit theorem, it is reasonable to assume that in general the noise distribution will be Gaussian in nature.

B. SOUND ABSORPTION IN SEA WATER

Losses of energy in a sound wave can be grouped into two types, spreading losses and attenuation losses. Spreading losses results from the simple geometric effect resulting from waves emitted from sources projecting in all directions. These kinds of losses, although great (spherical spreading losses proportional to $1/range^2$, cylindrical spreading losses proportional to $1/range$) do not affect the spectral characteristics of the ambient noise because all frequencies are equally attenuated by spreading losses. The major loss affecting the noise spectrum is the attenuation loss.

1. Absorption Coefficient of Sea Water

Any medium in which sound propagates is absorptive. This means that part of the sound wave energy is lost through heat as the wave propagates through the medium. By convention, the attenuation loss is identified as α and is expressed in dB per kilometer. What makes α so important in characterizing the ambient noise spectrum is that it has a complicated dependence on frequency. The complication arises because different loss processes dominate different portion of the frequency spectrum. As a result of these complications most of the current knowledge of the attenuation coefficient comes from measured data taken from various section of the ocean and from laboratory measurements. As a result of the complexity of the processes and of the changing nature of the oceans, no theoretical curves has been found to define the attenuation losses in a specific area, rather,

scientists have relied on curve fits over various parts of the frequency spectrum to define the magnitude of the attenuation coefficient. Three main effects cause the absorption of sound in sea water. The first is the result of shear viscosity. This effect was studied by Rayleigh who derived an expression equivalent to

$$\alpha = \frac{16\pi^2 \mu_s}{3\rho c^3} f^2, \tag{2.2}$$

where α is the intensity absorption coefficient, μ_s is the shear viscosity, ρ is the density, c is the sound velocity, and f is the frequency (Hz) [Ref. 6].

The value of the absorption coefficient calculated using this formula which considers only shear viscosity, is only about one third of the attenuation actually observed in pure distilled water. A second kind of viscosity called volume viscosity adds to the value of the absorption coefficient. This occurs as a result of the time lag required for the water molecules to flow into lattice interstices in the crystal structure. Because of this effect, the absorption coefficient formula becomes

$$\alpha = \frac{16\pi^2}{3\rho c^3}\left(\mu_s + \frac{3}{4}\mu_v\right) f^2, \tag{2.3}$$

where μ_v is the volume viscosity coefficient [Ref. 4].

Figure 2.4 presents three different curves for the value of the absorption coefficient as a function of frequency. The first curve shows the measured values for sea water taken from Urick [Ref. 4]. The second curve shows the measured absorption coefficient in distilled water which considers both the shear and the volume viscosity factors in accordance with Equation (2.3). The third shows the theoretical absorption using only the shear viscosity calculated with Equation (2.2).

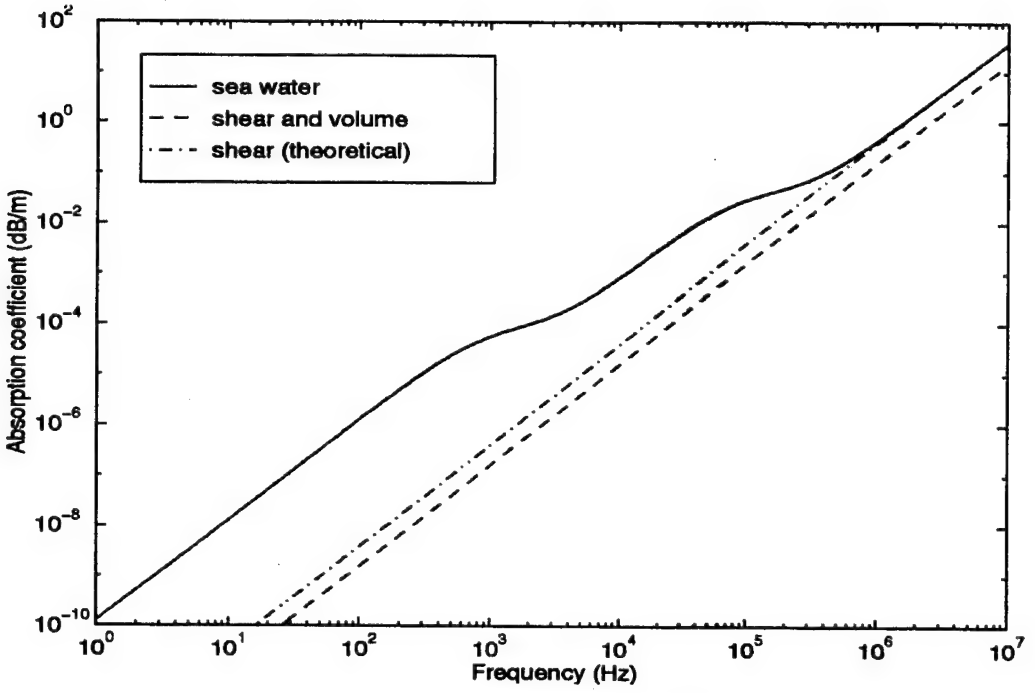

Figure 2.4: Comparison of absorption coefficients in sea water, due to shear and volume viscosity, and shear viscosity alone.

It is evident from the figure that not all absorption is accounted for by using only the effects of shear viscosity and volume viscosity. This is particularly true in frequencies below 100 kHz which are of particular interest for the sonar detection problems. For example, the absorption coefficient of sea water in the frequency range of 5 to 50 kHz is actually about 30 times that of distilled water [Ref. 4]. It has been found that in these frequencies the additional absorption mechanisms are due to the different salts dissolved in sea water. Leonard, Combs, and, Skidmore [Ref. 7] have shown that in addition to water, two other components of sea water have been found to be important to the strength of the absorption coefficient. These are magnesium sulfate ($MgSO_4$) and boric acid ($B(OH)_3$). In particular, in the frequencies of most interest for the detection problem, 10 to 5 kHz, the effect of both absorptive components is very important. The principal component of the salts dissolved in sea water, Sodium Chloride (NaCl), does not appear to be a significant absorptive constituent. Despite the fact that $MgSO_4$ constitutes only about 4.7 percent by

weight of the total dissolved salts in sea water, it has been found to be the dominant absorptive constituent in sea water for frequencies below 100 kHz [Ref. 7]. The absorption is due to the ionic relaxation of the $MgSO_4$ molecules in sea water. This is a process during which $MgSO_4$ ions dissociate and reassociate during a finite time interval, called the relaxation time, due to the pressure effect of a sound wave. Therefore, it is considered that sound absorption in sea water is the sum of the absorption due to water, magnesium sulfate, and boric acid.

Fisher and Simmons, in [Ref. 8] show that the absorptive coefficient also has a dependence on pressure. However since pressure does not vary with frequency, but rather with depth, its effect on the spectral shape of the ambient noise is not relevant to this study. Figure 2.5 shows the effect of magnesium sulfate and boric acid on the absorption coefficient of fresh water. These curves were derived from laboratory measurements [Ref. 8].

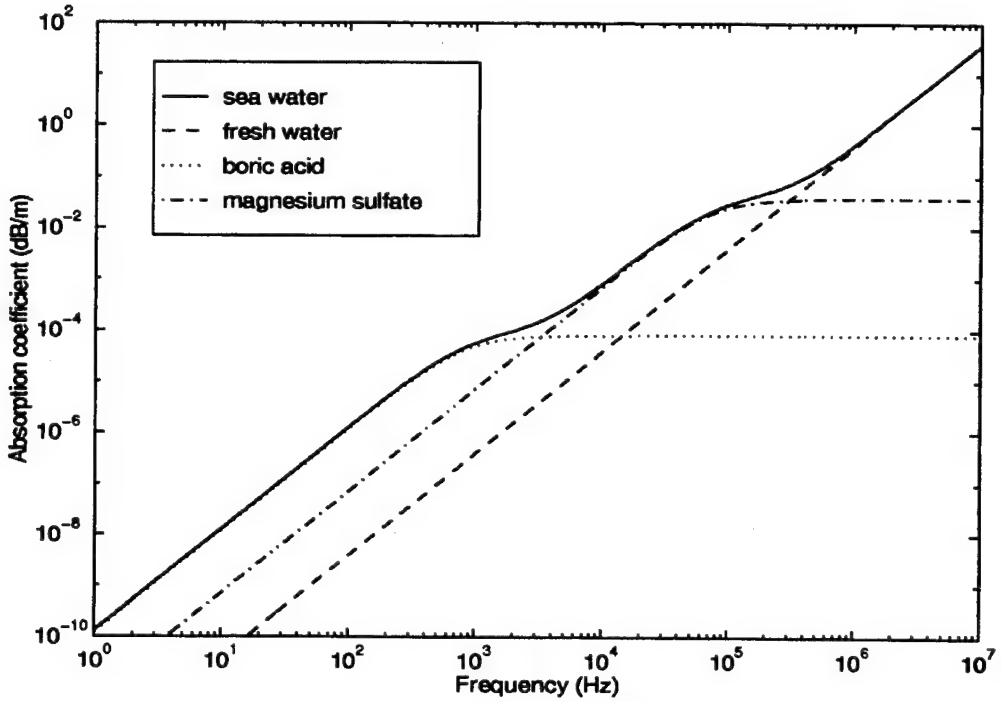

Figure 2.5: Sound absorption at 5°C, 1 atm, 35 ppt salinity, and pH=8.0. After Fisher and Simmons [Ref. 8].

2. Resonating Bubbles

Gas bubbles in the oceans can generate extremely high sound attenuation. Although the concentration of bubbles in the main body of the ocean is insignificant compared to the other sources of attenuation, breaking waves, schools of fishes with their internal air sacs, and, travelling ships and submarines often create large concentrations of gas bubbles. A sound wave passing through an area with many gas bubbles will lose energy as a result of the viscous forces and heat conduction generated by the compressive/decompressive effects of the wave. The presence of the gas bubbles affects the physical characteristics of the medium by changing its sound speed and density. This will cause reflective and refractive effects which will further dissipated the energy of the sound. Finally, since the gas bubbles in the water create an inhomogeneous medium, the sound wave will also be scattered, meaning that some energy will be redirected in all directions as a result of encountering a bubble. The scattering effects is largely dependent on the resonant frequency of the bubbles. The resonant frequency is given by

$$f_R = \frac{1}{2\pi a}\sqrt{\frac{3\gamma P_h}{\rho_o}} \quad , \tag{2.4}$$

where a is the bubble radius, γ is the ratio of specific heat for the gas in the bubble, P_h is the hydrostatic pressure at the depth of the bubble, and, ρ_o is the density of the surrounding medium [Ref. 1].

Frequencies in the acoustic wave at or close to the resonant frequencies of the bubbles can have great influence on the scattering effect while those frequencies which are further from the resonant frequency may not be scattered. Fish detectors can find schools of fish and determine the size of the fishes present by extrapolating their size from the dominant resonant frequency created by their air sacs. The measure of attenuation of a bubble is determined by its effective cross section σ. This value is a measure of the fraction of energy that the bubble will extract from a sound beam of 1 square meter cross section. If an

acoustic wave contains only frequencies at the resonant frequency f_R of the bubble, the effective cross section can be more than a thousand times greater that its actual cross section [Ref. 1]. When the frequencies of the acoustic wave are greater than the resonant frequency of the bubble, the effective cross section of the bubble affecting the wave is approximately the same as that of the bubbles. When the frequencies are smaller, the effective cross section is much less than the actual cross section of the bubble and quickly becomes negligible. Since bubbles rarely exist all with the same radius, those bubbles with the greatest scattering effect will be those whose resonant frequencies are closest to the frequencies contained in the acoustic wave. Since ambient noise contains all frequencies, the presence of bubbles in the area of interest will further change the frequency spectra of the ambient noise. However, there is a practical limit to the size of the bubbles in the ocean. Ocean conditions resulting in the creation of bubbles can be expected to generate bubbles with a distribution around one predominant bubble size. It can also be expected that bubbles can only attain a certain maximum size and therefore will only affect frequencies which are above the resonant frequency of the largest bubbles. Therefore, the ambient noise spectrum in regions affected by bubbles will have a maximum at the frequency associated with the predominant bubble size. Its shape will depend on the distribution of the bubbles size and the amplitude of oscillations in the affected frequencies.

C. TRANSDUCER EFFECTS

The particular environment in which a receiver is to used is of prime importance to its designers. For the design of transducers, the most important considerations arise from the physical properties of water. In addition to these considerations, transducers must be designed to withstand the effects of sea water, biological activity, and, often large hydrostatic pressures [Ref. 9].

The ultimate performance of a transducer is limited by the signal to noise ratio of its output. The noise level at this point is the result of the ambient noise in the ocean at the receiving location and of the additional noise caused by the transducer itself. Turbulent

boundary layer pressure fluctuations, mechanical vibrations, and, electrical noise, are but some of the possible causes of noise. Obviously the success of the design will be in large part dependent on how well these noise sources have been dealt with. Each of these causes may affect different portions of the frequency spectrum based on their inherent characteristics. Hydrophones also have an inherent sensitivity which vary with frequency. Many different materials are used in the design of modern transducers; quartz, piezoelectric polymers such as polyvinylidene (PVDF), composite ceramics, etc. Each of these materials exhibit different characteristics and will have different frequency responses. In the design of the transducer, these characteristics are taken into consideration and the material selected is usually used in a spectral region where its frequency response is relatively constant and its sensitivity adequate for its mission. When the frequency response is not constant, it must then be "equalized" by means of calibration curves derived experimentally by the user. Nevertheless, this experimental "equalization" necessarily results in some distortion of the received signal. Notwithstanding the material chosen, many transducers perform very poorly at very low frequency (< 10 Hz) often as a result of having to be shielded from large hydrostatic pressures.

D. DISCRETE TIME SAMPLING

The sampling process is the operation by which an analog signal is converted into a corresponding sequence of discrete time samples which are uniformly spaced in time. Tranducers are inherently analog sensors. The pressure information they convey must be processed by an acoustic processor. The modern acoustic processor is invariably a digital computation device, therefore requiring discretization of the captured underwater signals. The derivation of the sampling theorem is presented in most books on electrical engineering and will not be repeated here. The sampling theorem for band-limited signals of finite energy can be stated as follows [Ref. 8]:

"A band-limited signal of finite energy, which has no components higher than *W Hz*, is completely described by specifying the values of the signal at instants of time separated by less then *1/(2W)* seconds."

The sampling rate of *2W* samples per second is known as the Nyquist rate. The corollary of this theorem, is that if the signal to be sampled contains frequency components higher than *W* Hz then it will not be completely described by its sampled values. This will result in a distortion of the signal known as *aliasing*. This factor refers to the effect wherein components of the signal which are at a frequency above *W*, will instead appears at a different frequency in the spectrum of the sampled signal. The distortion caused by aliasing can be dealt with by two methods. Aliasing can be eliminated by sampling the signal at a greater frequency than the Nyquist rate of the highest frequency component present in the signal. However when frequency components are present with frequencies which are above the capabilities of the sampling system, an analog low-pass filter can be applied to the signal prior to sampling, to remove all frequencies above one half of the sampling rate. This last method is usually favored due the availability of good, inexpensive low-pass filters.

Nevertheless, this process necessarily changes the power spectrum of the noise. For example when white noise is corrupting the signal. This noise once filtered no longer has equal power for all frequencies, but rather becomes a band-pass white noise process which has constant power within a spectral region less than one half the sampling frequency and is zero outside. Since perfect filters do not exist in nature, the filter will also exhibit some distortion especially near its cut-off frequency which will affect the spectral representation of the noise being passed through it.

E. NOISE POWER SPECTRAL DENSITY

The ambient noise power spectral density is the result of each of the effects discussed above. Of course the main contribution to the shape of the ambient noise spectra is the attenuation of the sound waves by sea water. Nevertheless, the noise sources, their number, intensity, distribution and type, also have an important effect on the form of the spectral

shape of the ambient noise. Figure 2.6 summarizes the characteristics of the ambient noise spectrum (i.e., the terms spectrum and spectral density are used interchangeably) resulting from the noise sources and mechanism discussed in this chapter.

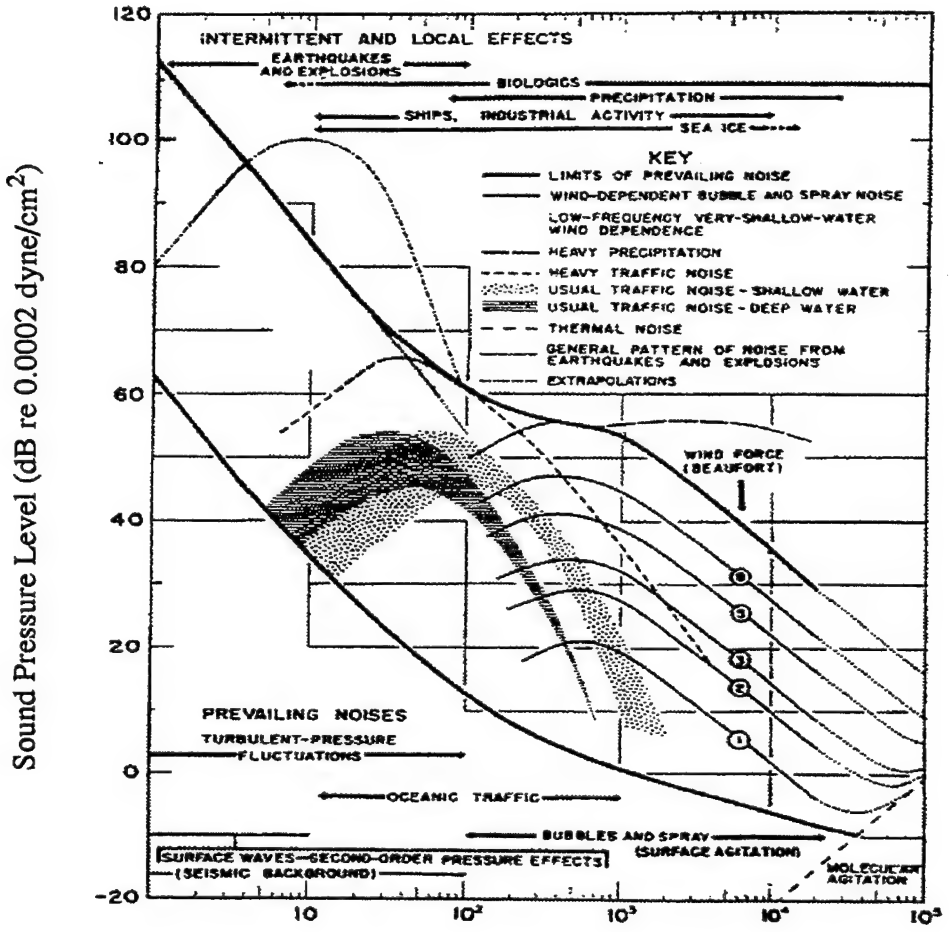

Figure 2.6: Ambient noise spectrum composite. From Ref. [3].

Figure 2.7 shows the ambient noise spectra of the ocean from the Point Sur SOSUS array. The signal was sampled at a frequency of 8 kHz resulting in a maximum frequency of 4 kHz.

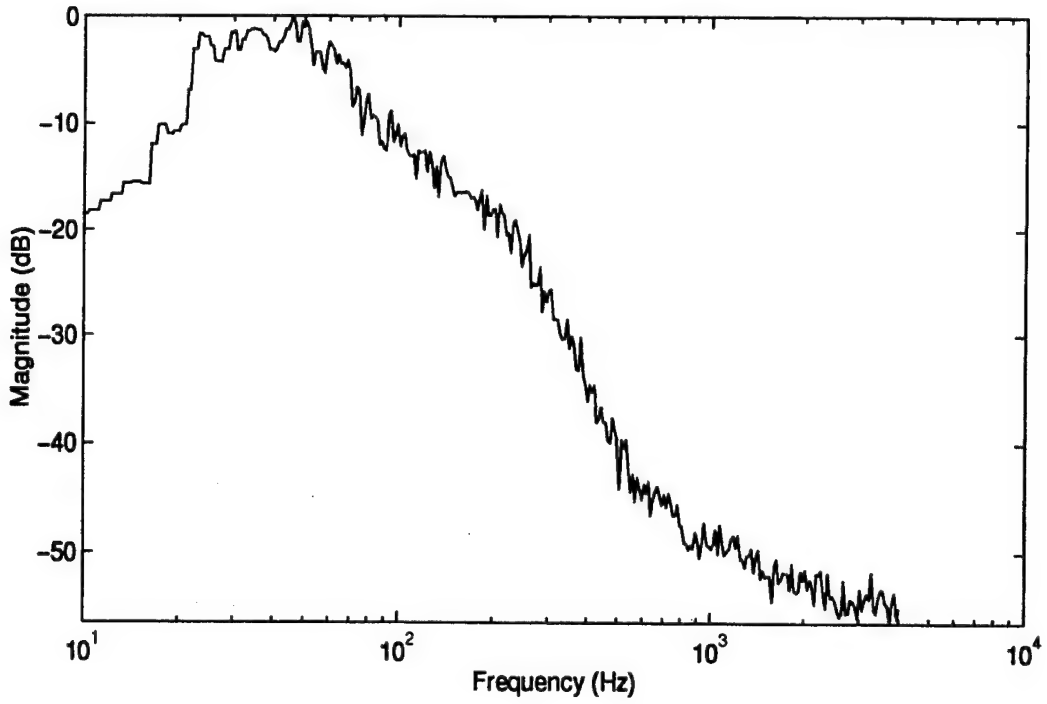

Figure 2.7: Normalized average sound pressure level of ambient noise at Point Sur SOSUS array (sampling frequency of 8 kHz)

As can be observed, the noise sound pressure level of the Point Sur array conforms to the general shape shown in Figure 2.6. At very low frequencies (10-100 Hz), shipping noise is clearly shown. At a frequency of approximately 200 Hz the characteristics hump caused by the addition of the wind dependent noise to the remaining high frequency components of the shipping noise is observed. At the higher frequencies shown on this figure, surface waves and agitation resulting from wind effects forms the general shape of the curve. In this case, the wind dependent noise appears to be low and the slope of the curve corresponds approximately to the lower limit of prevailing noise shown on Figure 2.6. From Figure 2.7, it can also be seen that in the frequency range shown, the slope drops-off approximately 52 dB for two decades or at a rate 26 dB/decade.

III. OPTIMAL DETECTION IN COLORED NOISE

The concept of optimal detection addresses the problem of conceiving the best detector at meeting one or several criteria given a number of assumptions. An example of a common criteria used for detection is the signal to noise ratio (SNR). Therefore, to be considered optimal in the sense of the SNR, a detector must provide the maximum SNR possible given the assumptions.

There are two general approaches in deriving an optimal detector for known signals in colored noise. Optimal solutions for signals in white noise are well known and attractive in their simplicity. Therefore, if a linear transformation can be applied to the received signal which whitens the noise while leaving the desired signal component intact, then this prefiltering operation can be combined with an optimal white noise detector as shown in Figure 3.1.

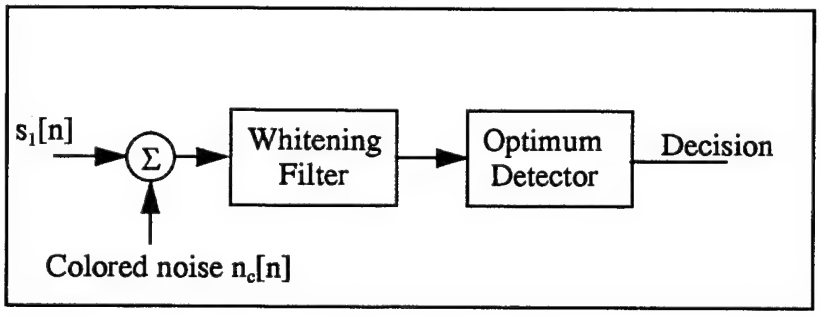

Figure 3.1: Optimal detection using whitening filter

Several methods are addressed in this chapter which can be used to whiten the colored noise. The noise is assumed to be Gaussian.

Alternatively, an optimal detector can also be derived directly in the form of a matched filter for colored noise. This filter has the advantage of removing the prewhitening step in the detection process and is shown in Figure 3.2.

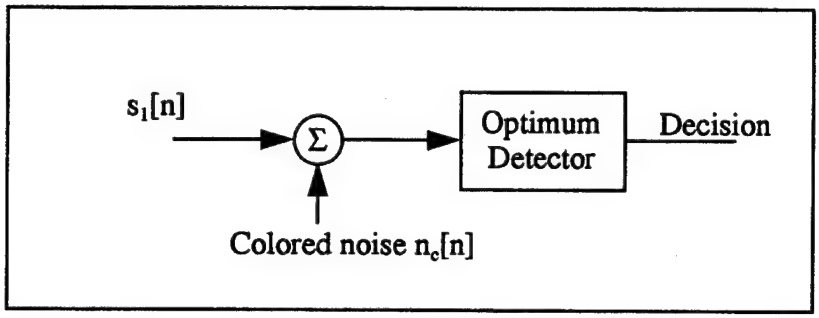

Figure 3.2: Optimal colored noise detector

A. DISCRETE TIME DETECTION

Since the first form of optimal detector for colored noise uses a white noise detector, it is first necessary to consider the case of discrete detection of known signals in additive white Gaussian noise (AWGN) [Ref. 12]. The power spectrum density of white Gaussian noise is a constant over the complete frequency range. Since we consider both negative and positive frequencies its magnitude is $No/2$. As the probability density function of a discrete random process is related to its autocorrelation by a Z-transform, the corresponding autocorrelation function for this probability density function is a delta function given by $N_o/2 \cdot \delta(n)$ [Ref. 12].

It is presumed that the aim of this exercise is to detect if a signal is present or not. The two hypotheses are given by:

H_0: $r[n] = n[n]$ Noise only,

H_1: $r[n] = s_1[n] + n[n]$ Signal s_1 plus noise.

In terms of matrix formulation, these are written as

H_0: $\boldsymbol{r} = \boldsymbol{n}$ Noise only,

H_1: $\boldsymbol{r} = \boldsymbol{s}_1 + \boldsymbol{n}$ Signal s_1 plus noise,

where the bold lower-case letters represent vectors. Bold upper-case letters are used to indicate that the symbol in question represents a matrix.

This is a binary detection scenario as shown in Figure 3.3.

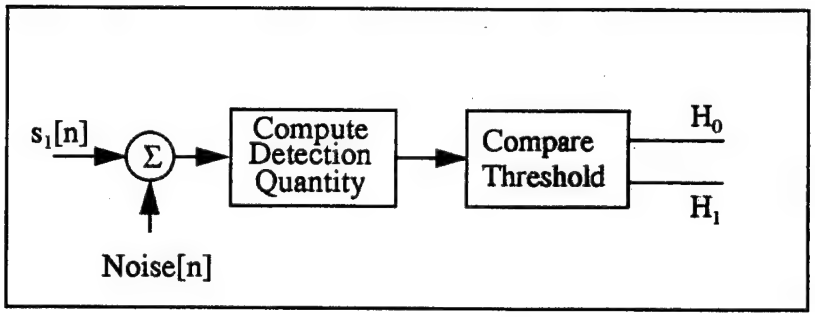

Figure 3.3: Binary detection

As shown in the figure, the optimum receiver computes the detection quantity, usually the likelihood ratio, and compares it to a threshold to decide which of the hypotheses provides the best answer. As we are dealing with discrete variables, the real life input (i.e., analog input) is sampled. As discussed in Chapter II, the sampling process changes the characteristics of the white noise to one that is bandlimited. As a result of this process, the correlation matrix of the noise is no longer a delta but rather takes the form of a digital sinc function.

In order to perform the detection using the standard likelihood ratio detector we require that the noise samples be uncorrelated. When the sampling frequency can be chosen, one way of ensuring that the noise samples are uncorrelated is to use a sampling procedure where the sampling interval corresponds to the distance from the origin to the first zero crossing of the noise correlation function. Since the noise samples are uncorrelated and Gaussian, they are independent. When data is provided with a set sampling frequency, the sampling interval can still be set by interpolation and resampling. The Gaussian nature of the samples allows completely statistical description using the first two moments.

Therefore, if the sampling interval is chosen to occur at the first zero crossing of the correlation matrix all noise samples are uncorrelated and the joint probability function between the signal and the noise component of the received discrete data can be expressed as a product of two independent terms. This then allows the use of a standard detector. Detectors designed to perform under the assumption of white noise are comprehensively addressed in [Refs. 10, 11, 12] and are not repeated here.

To devise the likelihood ratio detector, the mean and variance under each hypothesis must be calculated. Since the signals are known, the variance under each hypothesis is due only to the noise and is therefore the same. This is also the basis of the whitening property of the prediction error filter addressed later on in this chapter.

B. DETECTION IN COLORED NOISE

When the power spectral density of the noise is not a constant across the frequency range it is called colored noise. This means that the correlation matrix of the noise is not a delta function and the noise samples are correlated. In many signal processing applications, it is assumed that the noise samples are uncorrelated. Therefore it is often necessary to transform a vector of observations with correlated noise samples to one in which they are uncorrelated. As a result of such a transformation, the new correlation matrix of the observations is diagonal.

As is the case for white noise, the detection of a signal corrupted by colored noise is equivalent to determining which hypothesis is true. The hypotheses are

$$\begin{aligned} &H_0: \quad r[n] = n_c[n] && \text{Colored noise only,} \\ &H_1: \quad r[n] = s_1[n] + n_c[n] && \text{Signal } s_1 \text{ plus colored noise,} \end{aligned}$$

where r is the received signal, s_1 is the known signal and n_c is the Gaussian colored noise of zero mean, having a correlation matrix denoted by $\boldsymbol{R}_n$. Baye's detection formulation provides the general equation (i.e., Likelihood ratio function and threshold), which when using different detail information leads to a host of different detection criteria (i.e.,

maximum likelihood, min-max, maximum a posteriori, etc.). Basic detection theory for Gaussian colored noise is addressed in the following paragraph using the maximum a posteriori criterion.

1. Maximum A Posteriori Probability (MAP) Criterion

The detection technique whereas a decision must be made as to which one of binary or m-ary hypotheses is to be selected, is called hypothesis testing. In the case of target detection the two hypotheses are generally labelled as:

H_0: Target not present,

H_1: Target present.

For the detection process, a decision has to be made on the basis of observed random variables consisting of one or more observations. Various rules can be used for the detection and are derived by maximizing a measure of performance. The maximum a posteriori (MAP) decision rule usually leads to a partition of the decision space into two regions R_0 and R_1. To illustrate the MAP decision rule, we assume that a choice must be made between H_0 and H_1 based on one observation of the random variable Y where y is its actual value at the time of observation. The probability density functions of Y related to each hypothesis are known and are $f_{Y|H_0}(y|H_0)$ and $f_{Y|H_1}(y|H_1)$. The a posteriori probability that H_i is the correct hypothesis given the observation y is denoted by $P(H_i|y)$, for $i = 0, 1$. These are conditional probability density functions that are conditioned on the observation [Ref. 13]. The a priori probabilities will be denoted as $P(H_i)$, for $i = 0, 1$. These factors denote the known probabilities of each symbol being transmitted.

With Bayes rule, $P(H_i|y)$ can be written as

$$P(H_i|y) = \frac{f_{Y|H_i}(y|H_i)P(H_i)}{f_Y(y)}. \tag{3.1}$$

It is possible to maximize the probability of correct detection by deciding in favor of the hypothesis with the largest a posteriori probability;

Choose H_0 if $P(H_0|y) > P(H_1|y)$ or

choose H_1 if $P(H_1|y) > P(H_0|y)$.

These decision rules can also be written as

$$\frac{P(H_1|y)}{P(H_0|y)} \begin{array}{c} H_1 \\ > \\ < \\ H_0 \end{array} 1 \; . \tag{3.2}$$

With Equation (3.1) this becomes:

$$\frac{f_{Y|H_1}(y|H_1)}{f_{Y|H_0}(y|H_0)} \begin{array}{c} H_1 \\ > \\ < \\ H_0 \end{array} \frac{P(H_0)}{P(H_1)} \; , \tag{3.3}$$

where the left-hand expression is called the *likelihood ratio*, and the right-hand term the *decision threshold*. The MAP decision rule then consists of comparing the likelihood ratio to the detection threshold. Since this decision is based on the conditional probabilities, four possibilities exists in making the determination:

$P(D_0|H_0)$: Choose H_0 when H_0 is true: No target, no detection.

$P(D_1|H_1)$: Choose H_1 when H_1 is true: Target detected.

$P(D_0|H_1)$: Choose H_0 when H_1 is true: Target not detected.

$P(D_1|H_0)$: Choose H_1 when H_0 is true: False alarm.

where the $P(D_i|H_i)$ represents the conditional probability that decision D_i was taken given that H_i is the correct hypothesis.

The first two probabilities denote correct decisions with $P(D_1|H_1)$ identified as the *probability of detection* (P_D). $P(D_1|H_0)$ is called the *false alarm probability* (P_{FA}) and

$P(D_1|H_0)$ *probability of a miss* (P_M). Each of these regions is shown in Figure 3.4, where $f_1(y)$ denotes $f_{Y|H_1}(y|H_1)$ and $f_0(y)$ denotes $f_{Y|H_0}(y|H_0)$.

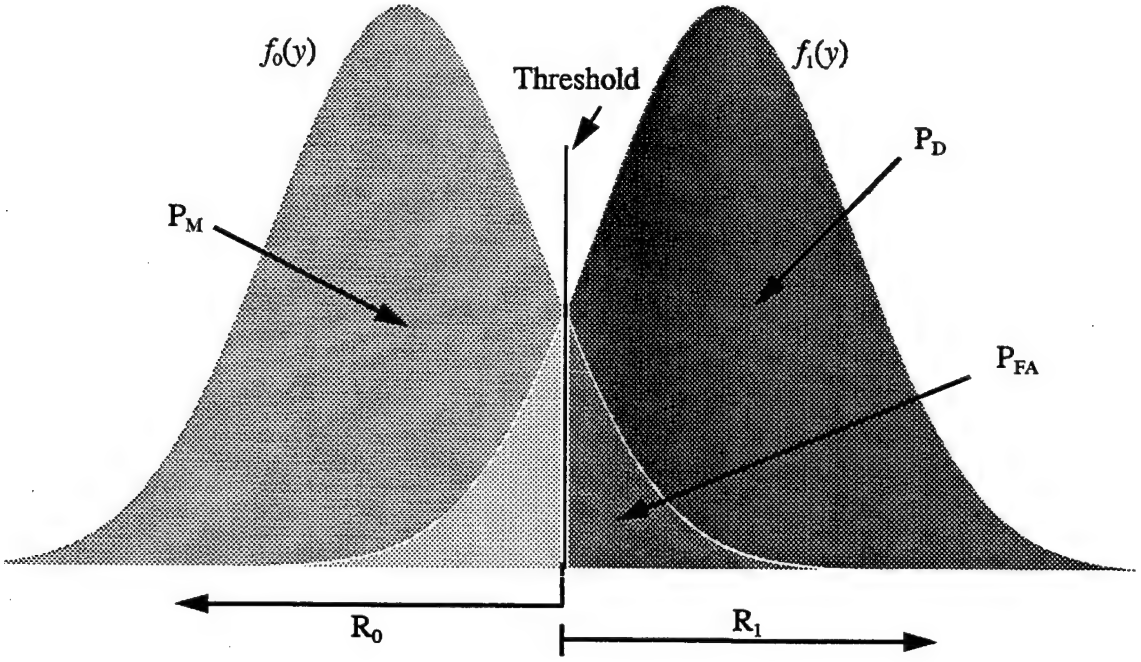

Figure 3.4: Detection regions: Gaussian probability density functions

In the binary detection problem, the receiver makes its decision by observing a number of m samples and then arrives at a decision by comparing the information extracted to a decision threshold. The Gaussian colored noise case is different from the Gaussian white noise problem in that the samples of the colored noise are correlated. The noise has a non-constant power spectral density function $S_n(\omega)$. In that interval it is not constant and varies according to the ambient noise affecting the observations. The MAP decision rule is then:

$$\frac{f_{Y|H_1}(\mathbf{y}|H_1)}{f_{Y|H_0}(\mathbf{y}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P(H_0)}{P(H_1)} . \tag{3.4}$$

The multivariate conditional probability density functions are Gaussian with the following characteristics

$$E\{\boldsymbol{Y}|H_0\} = \boldsymbol{s_0}, \tag{3.5}$$

$$E\{\boldsymbol{Y}|H_1\} = \boldsymbol{s_1}, \tag{3.6}$$

and

$$E\{[\boldsymbol{Y} - \boldsymbol{s_i}][\boldsymbol{Y} - \boldsymbol{s_i}]^T \Big| H_i\} = \boldsymbol{R_n}; \quad i = 0, 1, \tag{3.7}$$

where $\boldsymbol{s_0} = \boldsymbol{0}$ and $\boldsymbol{R_n}$ is the noise autocorrelation matrix. The conditional density under each hypothesis is

$$f_{\mathbf{Y}|H_0}(\boldsymbol{y}|H_0) = \frac{1}{2\pi\sqrt{det(\boldsymbol{R_n})}} e^{-\frac{1}{2}\boldsymbol{y}^T\boldsymbol{R_n}^{-1}\boldsymbol{y}}, \tag{3.8}$$

and

$$f_{\mathbf{Y}|H_1}(\boldsymbol{y}|H_1) = \frac{1}{2\pi\sqrt{det(\boldsymbol{R_n})}} e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{s_1})^T\boldsymbol{R_n}^{-1}(\boldsymbol{y}-\boldsymbol{s_1})}. \tag{3.9}$$

Substituting these in Equation (3.4) and taking the natural logarithm on both sides, leads to

$$\left(\frac{1}{2}\boldsymbol{y}^T\boldsymbol{R_n}^{-1}\boldsymbol{y} - \frac{1}{2}(\boldsymbol{y}-\boldsymbol{s_1})^T\boldsymbol{R_n}^{-1}(\boldsymbol{y}-\boldsymbol{s_1})\right) \underset{H_0}{\overset{H_1}{\gtrless}} \ln\left(\frac{P(H_0)}{P(H_1)}\right). \tag{3.10}$$

Since $\boldsymbol{R_n}$ is an m x m Toeplitz matrix, it can be decomposed into m distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ and m orthonormal eigenvectors $\boldsymbol{e_1}, \boldsymbol{e_2}, \ldots, \boldsymbol{e_m}$ with the following properties:

$$e_k^{*T} e_l = \begin{cases} 1, & k = l \\ 0, & k \neq l \end{cases}, \tag{3.11}$$

$$\boldsymbol{R}_x \boldsymbol{e}_i = \lambda_i \boldsymbol{e}_i \quad \text{, and} \tag{3.12}$$

$$\boldsymbol{e}^{*T}_i \boldsymbol{R}_x \boldsymbol{e}_i = \lambda_i . \tag{3.13}$$

If we group the eigenvectors in a matrix as

$$\boldsymbol{E} = \begin{bmatrix} | & | & \dots & | \\ \boldsymbol{e}_1 & \boldsymbol{e}_2 & \dots & \boldsymbol{e}_m \\ | & | & \dots & | \end{bmatrix}, \tag{3.14}$$

and the eigenvalues as,

$$\boldsymbol{D}_\lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ . & . & . & \dots & . \\ . & . & . & \dots & . \\ . & . & . & \dots & . \\ 0 & 0 & 0 & \dots & \lambda_m \end{bmatrix}, \tag{3.15}$$

then

$$\boldsymbol{E}^{*T} \boldsymbol{E} = \boldsymbol{I} \tag{3.16}$$

and

$$\boldsymbol{E}^{*T} \boldsymbol{R}_n \boldsymbol{E} = \boldsymbol{D}_\lambda . \tag{3.17}$$

Substituting Equations (3.11), (3.12), (3.13), (3.16), and, (3.17) into Equation (3.10) results in

$$\sum_{k=1}^{m} \frac{y_k' s'_{1,k}}{\lambda_k} \begin{matrix} H_1 \\ > \\ < \\ H_0 \end{matrix} \ln\left(\frac{P(H_0)}{P(H_1)}\right) + \frac{1}{2}\sum_{k=1}^{m} \frac{s'_{1,k}}{\lambda_k}, \tag{3.18}$$

where the apostrophe indicates that an orthogonalization transformation is required given by;

$$s'_{1,k} = e_k^{*T} s_1, \tag{3.19}$$

and

$$y'_{0,k} = e_k^{*T} y. \tag{3.20}$$

These transformations change the colored noise detection problem to an equivalent white noise detection problem. The decision rule shown in Equation (3.18), is similar to the white noise scheme with one exception. In the case of white noise, the noise components of the observations where assumed to have equal variance. In the case of colored noise, the orthogonal components of the noise do not have equal variance and this difference requires some normalization which is accomplished in the decision rule by normalizing with the eigenvalue associated with each eigenvector.

2. Basis of Whitening Transformations

In the following sections, several whitening transformations are presented. It is shown that by using various characteristics of a generic process x, i.e., power spectral density or correlation matrix, that we can whiten the power spectral density of x. This transformation is identified as

$$y = Ax, \tag{3.21}$$

where A is the whitening linear transformation. If x is colored noise only (i.e., H_0 is true), the transformation can be written as

$$n_w = A_{n_c} n_c, \tag{3.22}$$

where n_c and n_w are colored and white noise respectively. If x consists of signal plus noise we get

$$y = A_{n_c}(s + n_c), \tag{3.23}$$

or

$$y = A_{n_c}s + n_w. \tag{3.24}$$

The sequence y now consists of a transformed version of the signal and white noise.

Therefore the goal becomes finding a matrix A_{n_c} such that the noise is whitened. There are two fundamental ways in which a random vector with correlated components can be transformed into one with uncorrelated components. Because the correlation matrix of such a process is necessarily diagonal, this operation is often called diagonalization of the correlation matrix [Ref. 14].

The first method is through the use of an eigenvector factorization of the correlation matrix. The eigenvector factorization leads to a diagonal matrix of eigenvalues and a matrix of eigenvectors. We note that by definition the eigenvectors are orthogonal. The second method uses decompositions of the correlation, covariance, or data matrix by triangular factorization.

Since we are dealing with optimal discrete time detection, the next section starts with the derivation of the discrete matched filter. The general case where the correlation matrix of the noise and the signal are known is derived. It is shown that a solution to the matched filter problem is an eigenvalue/eigenvector problem.

C. DISCRETE TIME MATCHED FILTER

1. Deterministic Signal in Colored Noise

Let $x[n]$ be a finite length discrete vector of data to be processed. It consists of a deterministic signal $s[n]$ corrupted by additive colored noise $n_c[n]$:

$$x[n] = s[n] + n_c[n]. \tag{3.25}$$

The goal is to design a Finite Impulse Response (FIR) filter to optimally filter (in the sens of the matched filter) the data. We want the filter to pass the signal while rejecting the noise. The block diagram is shown in Figure 3.5,

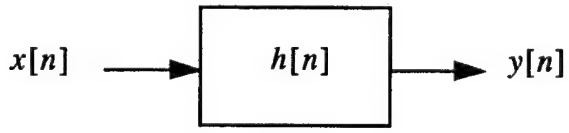

Figure 3.5: Matched filter

where $h[n]$ represents the filter impulse response and $y[n]$ is the result of the filtering. The output is

$$y[n] = \{s[n] + n_c[n]\} \otimes h[n] = y_s[n] + y_n[n], \tag{3.26}$$

where the symbol $\otimes$ identifies a convolution, $y_s[n]$ is the response due to the signal and $y_n[n]$ is the response due to the noise. Obviously to be successful we must maximize the former while trying to minimize the latter. Therefore, the filter operation is to maximize the output signal to noise ratio.

An optimal solution can be obtained by the following derivation maximizing the SNR [Ref. 14]. By definition,

$$SNR \equiv \frac{|y_s[n]|^2}{E\{|y_n[n]|^2\}}. \tag{3.27}$$

Since the duration of the signal is of length P, it is assumed that the filter is of a length sufficient to allows it to process the complete signal. We define the following four vectors:

$$\boldsymbol{s} = \begin{bmatrix} s[n_0] \\ s[n_1] \\ \\ . \\ . \\ . \\ s[n_{P-1}] \end{bmatrix}, \tag{3.28}$$

$$\boldsymbol{n_c} = \begin{bmatrix} n_c[n_0] \\ n_c[n_1] \\ \\ . \\ . \\ . \\ n_c[n_{P-1}] \end{bmatrix}, \tag{3.29}$$

$$\boldsymbol{x} = \boldsymbol{s} + \boldsymbol{n_c} = \begin{bmatrix} x[n_0] \\ x[n_1] \\ \\ . \\ . \\ . \\ x[n_{P-1}] \end{bmatrix}, \tag{3.30}$$

and

$$\boldsymbol{h} = \begin{bmatrix} h[n_0] \\ h[n_1] \\ \\ . \\ . \\ . \\ h[n_{P-1}] \end{bmatrix}. \tag{3.31}$$

From Equation (3.26) and the definition of convolution,

$$y[n_{P-1}] = \sum_{k=0}^{P-1} h[k]x[n_{P-1} - k] = \boldsymbol{h}^T \tilde{\boldsymbol{x}} \ , \tag{3.32}$$

$$y_s[n_{P-1}] = \sum_{k=0}^{P-1} h[k]s[n_{P-1} - k] = \boldsymbol{h}^T \tilde{\boldsymbol{s}} \ , \tag{3.33}$$

and

$$y_n[n_{P-1}] = \sum_{k=0}^{P-1} h[k]n_c[n_{P-1} - k] = \boldsymbol{h}^T \tilde{\boldsymbol{n}}_c \ , \tag{3.34}$$

where the ~ indicates time reversal of the vector. With the last two expressions, the SNR given by Equation (3.27) can be expressed as

$$SNR = \frac{(\boldsymbol{h}^{*T} \tilde{\boldsymbol{s}})^* (\tilde{\boldsymbol{s}}^T \boldsymbol{h})}{E[(\boldsymbol{h}^T \tilde{\boldsymbol{n}}_c)^* (\tilde{\boldsymbol{n}}_c^T \boldsymbol{h})]} \tag{3.35}$$

or

$$SNR = \frac{\boldsymbol{h}^{*T} \tilde{\boldsymbol{s}}^* \tilde{\boldsymbol{s}}^T \boldsymbol{h}}{\boldsymbol{h}^{*T} \boldsymbol{R}_n \boldsymbol{h}}. \tag{3.36}$$

In order to perform the optimal detection, we must find the impulse response $\boldsymbol{h}$ which maximizes the SNR. If we set $\boldsymbol{h}^{*T}\boldsymbol{R}_n\boldsymbol{h} = 1$, we can concentrate on maximizing the numerator. This does not affect our answer since this condition only serves to normalize the value of the impulse response. Therefore, the signal-to-noise ratio becomes

$$SNR = \boldsymbol{h}^{*T}\tilde{\boldsymbol{s}}^{*}\tilde{\boldsymbol{s}}^{T}\boldsymbol{h}. \tag{3.37}$$

The vector $\boldsymbol{s}$ is the signal, so our task is to find the correct $\boldsymbol{h}$. We can do this by the use of the Lagrange multiplier [Ref. 14]. We want to maximize the quantity

$$L = \boldsymbol{h}^{*T}\tilde{\boldsymbol{s}}^{*}\tilde{\boldsymbol{s}}^{T}\boldsymbol{h} + \lambda(1 - \boldsymbol{h}^{*T}\boldsymbol{R}_n\boldsymbol{h}). \tag{3.38}$$

In this case, since the constraint is real, λ is a real Lagrange multiplier. To find the minimum we must then find the vector $\boldsymbol{h}$ for which the gradient of L with respect to $\boldsymbol{h}^{*}$ is zero;

$$\nabla_{\boldsymbol{h}^{*}}L = \tilde{\boldsymbol{s}}^{*}\tilde{\boldsymbol{s}}^{T}\boldsymbol{h} - \lambda\boldsymbol{R}_n\boldsymbol{h} = 0. \tag{3.39}$$

By restating the equality we get

$$(\tilde{\boldsymbol{s}}^{*}\tilde{\boldsymbol{s}}^{T})\boldsymbol{h} = \lambda\boldsymbol{R}_n\boldsymbol{h}, \tag{3.40}$$

which is a generalized eigenvalue problem. It follows from this that $\boldsymbol{h}$ must be a generalized eigenvector of Equation (3.40). Since $\boldsymbol{h}$ is a solution to Equation (3.40), and by using $\boldsymbol{h}^{*T}\boldsymbol{R}_n\boldsymbol{h} = 1$, then

$$\boldsymbol{h}^{*T}\tilde{\boldsymbol{s}}^{*}\tilde{\boldsymbol{s}}^{T}\boldsymbol{h} = \lambda\boldsymbol{h}^{*T}\boldsymbol{R}_n\boldsymbol{h} = \lambda. \tag{3.41}$$

This last result shows that the SNR related to the generalized eigenvector $\boldsymbol{h}$ is equal to the eigenvalue λ corresponding to that eigenvector.

It can be shown that the matrix $\tilde{\boldsymbol{s}}^{*}\tilde{\boldsymbol{s}}^{T}$ is of unit rank [Ref. 14], therefore it is possible to find P-1 linearly independent vectors each of which is orthogonal to the vector $\tilde{\boldsymbol{s}}^{*}$ of

length P. Since each of these independent vectors are also orthogonal to $\tilde{s}^*$, each must be an eigenvector with a corresponding eigenvalue of zero. Therefore, the resulting SNR related to these eigenvectors is also zero. We will show next that the eigenvector related to the largest eigenvalue is proportional to $\boldsymbol{R}_n^{-1}\tilde{s}^*$ [Ref. 14: p. 243]. This is done by transforming the generalized eigen-decomposition problem of Equation (3.40) to an ordinary eigen-decomposition problem.

If the matrix $\boldsymbol{R}_n$ is expanded in terms of its eigenvalue and eigenvector matrices

$$\boldsymbol{R}_n = \boldsymbol{E}_n \boldsymbol{D}_n \boldsymbol{E}_n^{*T}, \tag{3.42}$$

and we let

$$\boldsymbol{h} = \boldsymbol{E}_n^{*T^{-1}} \boldsymbol{D}_n^{-1/2} \boldsymbol{g}, \tag{3.43}$$

and

$$\tilde{s}^* = \boldsymbol{E}_n \boldsymbol{D}_n^{1/2} \boldsymbol{t}. \tag{3.44}$$

Inserting the last three Equations into Equation (3.40) results in

$$(\boldsymbol{t}\boldsymbol{t}^{*T})\boldsymbol{g} = \lambda \boldsymbol{g} \tag{3.45}$$

which is an ordinary eigenvalue problem. It can also be demonstrated that the matrix $(\boldsymbol{t}\boldsymbol{t}^{*T})$ is of unit rank and that it also has only one eigenvector with a corresponding eigenvalue not equal to zero. This eigenvector must be proportional to $\boldsymbol{t}$. Therefore

$$\boldsymbol{g} \propto \boldsymbol{t}. \tag{3.46}$$

Inserting Equations (3.43) and (3.44) into (3.46) yields

$$\boldsymbol{D}_n^{1/2} \boldsymbol{E}_n^{*T} \boldsymbol{h} \propto \boldsymbol{D}_n^{-1/2} \boldsymbol{E}_n^{-1} \tilde{s}^*, \tag{3.47}$$

which when solved for $\boldsymbol{h}$ results in

$$\boldsymbol{h} \propto (\boldsymbol{E}_n^{*T})^{-1} \boldsymbol{D}_n^{-1} (\boldsymbol{E}_n)^{-1} \tilde{s}^*. \tag{3.48}$$

Since $\boldsymbol{E}$ is orthonormal, $\boldsymbol{E}^{-1} = \boldsymbol{E}^{*T}$, and

$$\boldsymbol{h} \propto \boldsymbol{E}_n \boldsymbol{D}_n^{-1} \boldsymbol{E}_n{}^{*T} \tilde{\boldsymbol{s}}^*, \tag{3.49}$$

or

$$\boldsymbol{h} \propto \boldsymbol{R}_n^{-1} \tilde{\boldsymbol{s}}^* \ . \tag{3.50}$$

We can also show that $\boldsymbol{R}_n^{-1}\tilde{\boldsymbol{s}}^*$ is proportional to the correct eigenvector by inserting it in Equation (3.40) which then results in

$$\tilde{\boldsymbol{s}}^* \tilde{\boldsymbol{s}}^T \boldsymbol{R}_n^{-1} \tilde{\boldsymbol{s}}^* = \lambda \boldsymbol{R}_n \boldsymbol{R}_n^{-1} \tilde{\boldsymbol{s}}^* = \lambda \tilde{\boldsymbol{s}}^* . \tag{3.51}$$

Since the equality stands, this shows that our assumption is correct and that $\tilde{\boldsymbol{s}}^T \boldsymbol{R}_n^{-1} \tilde{\boldsymbol{s}}^*$ is a scalar equal to the eigenvalue λ as shown in the next equation:

$$\tilde{\boldsymbol{s}}^* (\tilde{\boldsymbol{s}}^T \boldsymbol{R}_n^{-1} \tilde{\boldsymbol{s}}^*) = \tilde{\boldsymbol{s}}^* \lambda . \tag{3.52}$$

Because $\boldsymbol{R}_n$ is Toeplitz, $\tilde{\boldsymbol{s}}^T \boldsymbol{R}_n^{-1} \tilde{\boldsymbol{s}}^*$ can be rewritten as $\boldsymbol{s}^{*T} \boldsymbol{R}_{\mathrm{n}}^{-1} \mathrm{s}$. Therefore, we can write the maximum SNR as

$$SNR_{MAX} = \lambda_{MAX} = \boldsymbol{s}^{*T} \boldsymbol{R}_n^{-1} \boldsymbol{s} . \tag{3.53}$$

We now know that

$$\boldsymbol{h} = \alpha \boldsymbol{R}_n^{-1} \tilde{\boldsymbol{s}}^* , \tag{3.54}$$

where α is the normalizing factor that remains to be found. Inserting Equation (3.54) in the condition $\boldsymbol{h}^{*T} \boldsymbol{R}_n \boldsymbol{h} = 1$ and solving for α gives

$$\alpha = \frac{1}{\sqrt{\boldsymbol{s}^{*T} \boldsymbol{R}_n^{-1} \boldsymbol{s}}} . \tag{3.55}$$

Using Equations (3.54) and (3.55) the optimal filter is then defined as

$$\boldsymbol{h} = \frac{1}{\sqrt{\boldsymbol{s}^{*T} \boldsymbol{R}_n^{-1} \boldsymbol{s}}} \boldsymbol{R}_n^{-1} \tilde{\boldsymbol{s}}^* . \tag{3.56}$$

This solution can be interpreted as applying a whitening transformation to the sequence and using the reversed replica of the transformed known signal to achieve optimal detection.

If the noise is white, its correlation matrix is a diagonal matrix with diagonal elements σ_n^2. In this case Equation (3.40) becomes

$$(\tilde{\boldsymbol{s}}^* \tilde{\boldsymbol{s}}^T)\boldsymbol{h} = \lambda \sigma_n^2 \boldsymbol{h} \ , \tag{3.57}$$

which is an ordinary eigen-decomposition problem.

2. Random Signal in Colored Noise

The derivation of the previous section for a deterministic signal can be extended to that of random signals [Ref. 14]. Let the signal $s[n]$ and colored noise $n_c[n]$ be random processes. As before, it is assumed that the signal and the noise are independent, that we have P samples, and that the signal is present in all samples. Starting this time from the signal to noise ratio for two random processes [Ref. 14]:

$$SNR \equiv \frac{E\{|y_s[n_p]|^2\}}{E\{|y_n[n_p]|^2\}}, \tag{3.58}$$

where

$$E\{|y_s[n_p]|^2\} = \boldsymbol{h}^{*T} \boldsymbol{R}_s \boldsymbol{h} \tag{3.59}$$

and

$$E\{|y_n[n_p]|^2\} = \boldsymbol{h}^{*T} \boldsymbol{R}_n \boldsymbol{h}, \tag{3.60}$$

and, $\boldsymbol{R}_s$ and $\boldsymbol{R}_n$ are the correlations matrices of the signal and the noise respectively. As before, the magnitude of $\mathbf{h}$ is constrained so that $\boldsymbol{h}^{*T} \boldsymbol{R}_n \boldsymbol{h} = 1$. The SNR relation becomes

$$SNR = E\{|y_s[n_p]|^2\} = \boldsymbol{h}^{*T} \boldsymbol{R}_s \boldsymbol{h}. \tag{3.61}$$

Using the Lagrange multiplier again, to maximize the quantity L;

$$L = \boldsymbol{h}^{*T}\boldsymbol{R}_s\boldsymbol{h} + \lambda(1 - \boldsymbol{h}^{*T}\boldsymbol{R}_n\boldsymbol{h}). \tag{3.62}$$

To find the minimum we must then find the vector $\boldsymbol{h}$ for which the gradient of L with respect to $\boldsymbol{h}$ is zero:

$$\nabla_{\boldsymbol{h}^*}L = \boldsymbol{R}_s\boldsymbol{h} - \lambda\boldsymbol{R}_n\boldsymbol{h} = 0. \tag{3.63}$$

By restating the equality we get

$$\boldsymbol{R}_s\boldsymbol{h} = \lambda\boldsymbol{R}_n\boldsymbol{h}, \tag{3.64}$$

which is a generalized eigenvalue problem which **h** must satisfy. From this it can be concluded that as is seen in the deterministic case, the matched filter corresponds to the generalized eigenvector ($\boldsymbol{h}$) with the largest eigenvalue. Since $\boldsymbol{R}_s$ cannot be decomposed into its components when $s[n]$ is random, a simple expression for $\boldsymbol{h}$ cannot be given analytically and the generalized eigenvalue problem must be solved numerically to find the generalized eigenvector relating to the largest eigenvalue.

D. INVERSE FILTER

In detection problems, the goal is to detect the presence or absence of the signal. In underwater acoustic applications, the data consists of the signal(s) of interest which must be detected, and the sum total of all transformations and noises added by the ocean and the transducer. These effects can be viewed as a channel and can be modeled as a linear system. Maximization of the probability of detection of the signal requires an inverse filtering operation. If a perfect inversion process can be designed, the resultant signal from the inverse system will recover the original signal. In underwater acoustics, the major effects are due to the colored noise and the sea water attenuation. Therefore, the solution to the detection problem is to design a corrective system which when applied to the data, minimizes the effects of the colored noise and yield a replica of the original signal embedded in white (residual) noise. Then a standard detector can be used to obtain optimal

detection. In linear systems theory [Ref. 15], this corrective system is called an inverse system. In communication systems, it is called an equalizer. This inverse system will have a frequency response which is the reciprocal of the distortions resulting from the channel. This results in the use of Least-squares filter design methods. In our case, the objective is to remove the effects of attenuation and of colored noise sources to ensure that the power spectral density of the additive noise corrupting the desired signal is white.

A colored noise process $n_c[n]$ can be represented by a white noise process $n_w[n]$ transformed by a linear filter that is causally invertible [Ref. 16] as shown in Figure 3.6.

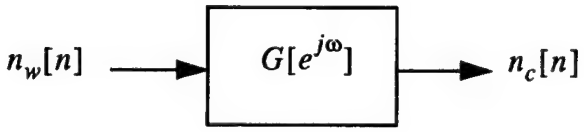

Figure 3.6: Colored noise created by linear invertible filter $G[e^{j\omega}]$.

The power spectral density of the colored noise becomes

$$S_c(\omega) = \sigma_w^2 \left| G[e^{j\omega}] \right|^2. \tag{3.65}$$

Since the frequency response $G[e^{j\omega}]$ is invertible,

$$H[e^{j\omega}] = \frac{1}{G[e^{j\omega}]}. \tag{3.66}$$

$S_c(\omega)$ becomes

$$S_c(\omega) = \frac{\sigma_w^2}{\left| H[e^{j\omega}] \right|^2}. \tag{3.67}$$

If the noise is to be whitened and the correct power spectral density (σ_w^2) is achieved, the inverse filtering operation must take place as shown in Figure 3.7 where the noise $n_c[n]$ is whitened by the transformation $H[e^{j\omega}]$.

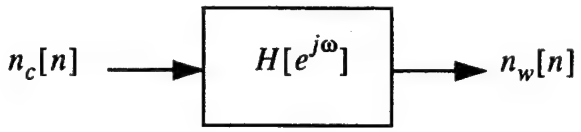

Figure 3.7: Colored noise whitened by inverse filter $H[e^{j\omega}]$.

If it is assumed that the power spectral density of the colored noise $S_c(\omega)$ is known, the whitening filter $H[e^{j\omega}]$ can be designed. If a signal $s[n]$ is present in the colored noise, the transformation will also affect it. However, because it is a linear transformation, the frequency of the signal is not affected. Since only the power spectral density of the colored noise is used in the design of the filter, the ratio of the spectral heights of the signal and noise at the signal frequency is preserved and the transformed noise is whitened.

E. WHITENING BY UNITARY TRANSFORMATION

1. Eigenvector Factorization

Since the eigenvectors of the correlation matrix for the random vector **x** are orthogonal, they can be used to perform a unitary transformation which whitens, hence orthogonalizes, the noise components of **x**. To elaborate on this method, let $\boldsymbol{R}_x$ be the correlation matrix of generic random vector $\boldsymbol{x}$. $\boldsymbol{R}_x$ is then related to its eigenvalues λ_l and eigenvectors $\mathbf{e}_l$ in the usual way

$$\boldsymbol{R}_x \boldsymbol{e}_l = \lambda_l \boldsymbol{e}_l , \tag{3.68}$$

which means that the correlation matrix multiplied by one of its eigenvector is equal to the same eigenvector multiplied by the corresponding eigenvalue. Since $\boldsymbol{R}_x$ (N X N) is hermitian symmetric, we can find N orthonormal eigenvectors $\boldsymbol{e}_l$, for $l = 1 \ldots N$, each of which has a corresponding real valued eigenvalue λ_l, for $l = 1 \ldots N$ [Ref. 14]. Since the eigen-

vectors are orthonormal, they have the property

$$e_k^{*T} e_l = \begin{cases} 1, & k = l \\ 0, & k \neq l \end{cases}. \tag{3.69}$$

Premultiplying Equation (3.68) by e_k^{*T} results in

$$e_k^{*T} R_x e_l = e_k^{*T} \lambda_l e_l, \tag{3.70}$$

and, since λ_l is a constant,

$$e_k^{*T} R_x e_l = \lambda_l e_k^{*T} e_l. \tag{3.71}$$

We known that $\mathbf{e}_k$ and $\mathbf{e}_l$ are orthonormal from Equation (3.69), then (3.71) becomes

$$e_k^{*T} R_x e_l = \lambda_l \cdot \begin{cases} 1, & k = l \\ 0, & k \neq l \end{cases}. \tag{3.72}$$

If we group the eigenvectors column-wise in a matrix,

$$E = \begin{bmatrix} | & | & \dots & | \\ e_1 & e_2 & \dots & e_N \\ | & | & \dots & | \end{bmatrix}, \tag{3.73}$$

then, since all the eigenvectors are orthonormal,

$$E^{*T} E = I. \tag{3.74}$$

We can now write Equation (3.72) as,

$$E^{*T} R_x E = D_\lambda, \tag{3.75}$$

where

$$\boldsymbol{D}_{\lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ | & | & | & | & | \\ 0 & 0 & 0 & \dots & \lambda_m \end{bmatrix}. \tag{3.76}$$

Let $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ be a linear transformation of the vector $\boldsymbol{x}$. Then,

$$E[\boldsymbol{y}] = E[\boldsymbol{A}\boldsymbol{x}] = \boldsymbol{A}E[\boldsymbol{x}]. \tag{3.77}$$

Therefore the mean of the new random vector $\boldsymbol{y}$ is a scaled version of the mean of $\boldsymbol{x}$.

For the correlation matrix,

$$\boldsymbol{R}_y = E[\boldsymbol{y}\boldsymbol{y}^{*T}] = E[\boldsymbol{A}\boldsymbol{x}\boldsymbol{x}^{*T}\boldsymbol{A}^{*T}], \tag{3.78}$$

$$\boldsymbol{R}_y = \mathrm{A}E[\boldsymbol{x}\boldsymbol{x}^{*T}]\mathrm{A}^{*T}, \tag{3.79}$$

and

$$\boldsymbol{R}_y = \boldsymbol{A}\boldsymbol{R}_x\boldsymbol{A}^{*T}. \tag{3.80}$$

Correspondingly the covariance matrix becomes

$$\boldsymbol{C}_y = \boldsymbol{A}\boldsymbol{C}_x\boldsymbol{A}^{*T}. \tag{3.81}$$

If we define the random vector $\mathbf{y}$ as

$$\boldsymbol{y} = \boldsymbol{E}^{*T}\boldsymbol{x}, \tag{3.82}$$

and we know from (3.75) that,

$$\boldsymbol{E}^{*T}\boldsymbol{R}_x\boldsymbol{E} = \boldsymbol{D}_{\lambda}, \tag{3.83}$$

then we can say using (3.82) that,

$$\boldsymbol{R}_y = \boldsymbol{E}^{*T}\boldsymbol{R}_x\boldsymbol{E} = \boldsymbol{D}_{\lambda} \tag{3.84}$$

where the matrix of eigenvectors $\boldsymbol{E}^{*T}$ of $\boldsymbol{R}_x$ produced the linear transformation of $\boldsymbol{x}$.

This shows that the correlation matrix of the new vector y is the matrix of eigenvalues of x which by definition is diagonal. This is an important transformation since the samples of y are uncorrelated. Since the elements of D_λ are always non-negative and usually positive, R_y is at least positive semidefinite and usually positive definite. This transformation whitens the samples of the random vector x.

If we premultiply Equation (3.84) by E and postmultiply by E^{*T}, given that $EE^{*T} = E^{*T}E = I$ because of their orthonormality, then

$$R_x = ED_\lambda E^{*T}. \tag{3.85}$$

This equation gives us the correlation matrix of x in terms of its eigenvectors and eigenvalues. Since E is orthonormal, $E^{-1} = E^{*T}$, and

$$R_x^{-1} = ED_\lambda^{-1}E^{*T}. \tag{3.86}$$

Since D_λ is a real element diagonal matrix with elements λ_i, for $i = 1...N$, D_λ^{-1}, is also a diagonal matrix but with elements $1/\lambda_i$, for $i = 1...N$. Since the λ_i of D_λ are all greater than zero when R_x is positive definite, the diagonal elements of D_λ also are greater than zero. Therefore, this equation provides us with a simple way to invert the correlation matrix.

2. Singular Value Decomposition

Eigenvector factorization works well when the correct correlation matrix of the process described by x is known. In practice however, the correlation matrix R_x must be estimated from a finite length of data or by an inverse Fourier transform of the estimated power spectral density. This may result in a matrix which is poorly conditioned or singular.

Since $\boldsymbol{R}_x$ is generally estimated, the eigenvalues and eigenvectors computed for the matrix may lack precision. A better way to compute the eigenvalues and eigenvectors of $\boldsymbol{R}_x$ is by using the singular value decomposition of the data matrix $\boldsymbol{X}$ (K x M) defined as

$$\boldsymbol{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_M \\ x_{M+1} & x_{M+2} & \cdots & x_{2M} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_{(K-1)M+1} & x_{(K-1)M+2} & \cdots & x_{KM} \end{bmatrix}. \tag{3.87}$$

The singular value decomposition theorem states that any K x M matrix $\boldsymbol{X}$ can be decomposed into the following product of matrices

$$\boldsymbol{X} = \boldsymbol{U}\Sigma\boldsymbol{V}^{*T}, \tag{3.88}$$

where $\boldsymbol{U}$ is K x K orthonormal matrix containing left singular vectors arranged column-wise,

$$\boldsymbol{U} = \begin{bmatrix} | & | & \cdots & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_K \\ | & | & \cdots & | \end{bmatrix}, \tag{3.89}$$

$\boldsymbol{V}$ is an M x M orthonormal matrix of right singular vectors,

$$\boldsymbol{V} = \begin{bmatrix} | & | & \cdots & | \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_M \\ | & | & \cdots & | \end{bmatrix}, \tag{3.90}$$

and Σ is a K x M matrix of non-negative real singular values

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ 0 & 0 & \dots & \sigma_M \\ 0 & 0 & \dots & 0 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ 0 & 0 & \dots & 0 \end{bmatrix} . \tag{3.91}$$

This last matrix is written for $K > M$. The σ_i are in decreasing order and may equal zero for the larger values of i. Generally there are r nonzero singular values σ where r is the rank of the matrix $\boldsymbol{X}$ (number of independent columns). If $K = M$ we have a diagonal matrix. When $K < M$ the matrix Σ has columns of zeros rather than having rows of zeros, and the rank r of $\boldsymbol{X}$ is equal to its number of independent rows. Whether the data is real or complex, the singular values are always real and greater than or equal to zero.

The equivalence of this technique to the eigenvalue-eigenvector factorization method is given as follows. By definition,

$$\boldsymbol{R}_x = \frac{1}{N}\boldsymbol{X}^{*T}\boldsymbol{X}, \tag{3.92}$$

and using Equation (3.88)

$$\boldsymbol{R}_x = \frac{1}{N}\boldsymbol{X}^{*T}\boldsymbol{X} = \frac{1}{N}\boldsymbol{V}\Sigma^T\boldsymbol{U}^{*T}\boldsymbol{U}\Sigma\boldsymbol{V}^{*T} = \boldsymbol{V}\left(\frac{1}{\sqrt{N}}\Sigma^T\Sigma\frac{1}{\sqrt{N}}\right)\boldsymbol{V}^{*T}, \tag{3.93}$$

$\boldsymbol{U}$ is orthonormal, that is $\boldsymbol{U}^{*T}\boldsymbol{U} = \boldsymbol{I}$, allowing the last step in Equation (3.93). This represents the factorization of $\boldsymbol{R}_x$ in the same form as the standard eigenvalue/eigen-

vector factorization,

$$\boldsymbol{R}_x = \boldsymbol{E}\boldsymbol{D}_\lambda \boldsymbol{E}^{*T}, \tag{3.94}$$

or by postmultiplying both sides by $\boldsymbol{E}$,

$$\boldsymbol{R}_x\boldsymbol{E} = \boldsymbol{E}\boldsymbol{D}_\lambda . \tag{3.95}$$

Since this decomposition is unique we can make the following statements: The matrix of right singular vectors $\boldsymbol{V}$ is the same as eigenvector matrix $\boldsymbol{E}$, and, the singular values of the matrix Σ are the square roots of the eigenvalues if the number of samples are accounted for.

Therefore the singular value decomposition can be used as an alternative way of finding the eigenvalues and eigenvectors of the correlation matrix of a function without having to estimate its correlation matrix. Because it provides a better evaluation of the eigenvalues and eigenvectors, the SVD is the method of choice for solving most linear-least-square problems. It provides a solution when other techniques such as Gaussian elimination or LU decomposition fail as a result of singular or close to singular matrices. The SVD can be done with a singular matrix, and provides a solution that is "almost" unique. Of course to be able to perform the SVD we first must have the data available to form the matrix, where the data must be signal free (i.e., noise only).

3. Mahalanobis Transformation

The Mahalanobis transformation is another transformation which can be used for whitening. It is also based on the eigendecomposition of the noise correlation matrix and is defined as follows [Ref. 14: p. 247]:

$$\boldsymbol{y} = \boldsymbol{E}\boldsymbol{D}_\lambda^{-1/2}\boldsymbol{E}^{*T}\boldsymbol{x} = \boldsymbol{R}_n^{-1/2}\boldsymbol{x} . \tag{3.96}$$

This transformation results in a diagonal correlation matrix of the random variable $\boldsymbol{y}$ as shown

$$R_y = ED_\lambda^{-1/2} E^{*T} R_x E D_\lambda^{-1/2} E^{*T} . \quad (3.97)$$

Inserting Equation (3.83) results in

$$R_y = ED_\lambda^{-1/2} D_\lambda D_\lambda^{-1/2} E^{*T} , \quad (3.98)$$

and

$$R_y = EIE^{*T} = I , \quad (3.99)$$

proving that the transformation shown in Equation (3.96) results in a unit variance white noise random process.

F. WHITENING BY TRIANGULAR FACTORIZATION

As previously discussed, transformations that whiten colored noise must result in a correlation matrix which is diagonal. Another way of achieving this effect is by triangular factorization, also known as triangularization. The term triangularization refers to the form of the matrix of orthogonal vectors which is used to orthogonalize the colored noise component of the received samples. Otherwise this technique is similar to whitening by a unitary transformation. This method leads to a transformation that can be interpreted as causal or anticausal linear filtering of the associated signal sequence depending on the use of the lower-upper triangularization or the upper-lower triangularization.

1. Lower-Upper Factorization (LDU)

A square matrix R_x can be expressed as

$$R_x = LD_L U , \quad (3.100)$$

where L is a unit lower triangular matrix, meaning that the diagonal elements are equal to 1, and all elements above the diagonal are equal to zero, D_L is a diagonal matrix, and U is a unit upper triangular matrix. Since the correlation matrix R_x is symmetric, this transformation takes the form

$$\boldsymbol{R}_x = \boldsymbol{L}\boldsymbol{D}_L\boldsymbol{L}^{*T}. \tag{3.101}$$

The correlation of a process can be diagonalized by the following transformation:

$$\boldsymbol{y} = \boldsymbol{L}^{-1}\boldsymbol{x}. \tag{3.102}$$

To show that this transformation diagonalizes the correlation matrix of $\boldsymbol{y}$ we begin with the correlation function for $\boldsymbol{y}$,

$$\boldsymbol{R}_y = E\{\boldsymbol{y}\boldsymbol{y}^{*T}\} \tag{3.103}$$

using Equation (3.102),

$$\boldsymbol{R}_y = E\left\{(\boldsymbol{L}^{-1}\boldsymbol{x})(\boldsymbol{L}^{-1}\boldsymbol{x})^{*T}\right\}, \tag{3.104}$$

$$= \boldsymbol{L}^{-1}E\{\boldsymbol{x}\boldsymbol{x}^{*T}\}(\boldsymbol{L}^{-1})^{*T}, \tag{3.105}$$

and,

$$= \boldsymbol{L}^{-1}\boldsymbol{R}_x(\boldsymbol{L}^{-1})^{*T}. \tag{3.106}$$

If Equation (3.101) is now inserted in Equation (3.106),

$$\boldsymbol{R}_y = \boldsymbol{L}^{-1}(\boldsymbol{L}\boldsymbol{D}_L\boldsymbol{L}^{*T})(\boldsymbol{L}^{-1})^{*T}. \tag{3.107}$$

Now $\boldsymbol{L}^{-1}\boldsymbol{L}$ and $\boldsymbol{L}^{*T}(\boldsymbol{L}^{-1})^{*T}$ cancel out and we are left with

$$\boldsymbol{R}_y = \boldsymbol{D}_L. \tag{3.108}$$

Since we know that $\boldsymbol{D}_L$ is a diagonal matrix this proves that the transformation does indeed diagonalize the correlation matrix of $\boldsymbol{y}$. If $\boldsymbol{R}_x$ is the correlation matrix of a colored noise process, then the transformation of Equation (3.102) whitens it.

When applied to a sequence of consecutive samples of a random sequence, this transformation is causal and results in a series of orthogonal random variables.

2. Upper-Lower Factorization (UDL)

An alternative triangular decomposition factors the correlation or covariance matrix into a upper-lower decomposition. This decomposition of a correlation matrix has the form

$$\boldsymbol{R}_x = \boldsymbol{U}\boldsymbol{D}_U\boldsymbol{U}^{*T}, \tag{3.109}$$

where $\boldsymbol{U}$ is unit upper triangular and the matrix $\boldsymbol{D}_U$ is diagonal. This last matrix is different than the matrix of the lower-upper decomposition.

If we premultiply the previous equation by $\boldsymbol{U}^{-1}$ and postmultiply by $(\boldsymbol{U}^{*T})^{-1} = (\boldsymbol{U}^{-1})^{*T}$,

$$\boldsymbol{U}^{-1}\boldsymbol{R}_x(\boldsymbol{U}^{-1})^{*T} = \boldsymbol{D}_U. \tag{3.110}$$

Therefore keeping in mind in mind the definition for $\boldsymbol{R}_x$, it can be shown that the transformation

$$\boldsymbol{y} = \boldsymbol{U}^{-1}\boldsymbol{x}, \tag{3.111}$$

also results in a vector with orthogonal components. Since $\boldsymbol{U}$ is upper triangular, this transformation is anti-causal. The matrix $\boldsymbol{U}$ can be computed with the LDU method in the following way [Ref. 17]:

Taking the reversal of Equation (3.109) results in

$$\tilde{\boldsymbol{R}}_x = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}_U\tilde{\boldsymbol{U}}^{*T}, \tag{3.112}$$

where the matrix $\tilde{\boldsymbol{U}}$ is lower diagonal. Equation (3.112) then has the LDU form of Equation (3.100). Since this decomposition is unique it means that the lower triangular matrix resulting LDU decomposition of the matrix $\tilde{\boldsymbol{R}}_x$ is equal to $\tilde{\boldsymbol{U}}$. The matrix $\boldsymbol{U}$ of Equation (3.111) is then found by reversing the lower triangular matrix achieved from $\tilde{\boldsymbol{R}}_x$.

3. Cholesky Factorization

The Cholesky decomposition is a special case of the LDU decomposition [Ref. 16]. If $\boldsymbol{R_x}$ can be expressed as

$$\boldsymbol{R_x = L D_L L^{*T}}, \tag{3.113}$$

and, the diagonal elements of $\boldsymbol{D_L}$ are positive, the matrix $\boldsymbol{R_x}$ can then be further expressed as

$$\boldsymbol{R_x = L_C L_C^{*T}}, \tag{3.114}$$

where $\boldsymbol{L_C}$ is lower triangular. Therefore the correlation matrix of a process can be diagonalized by the transformation

$$\boldsymbol{y = L_C^{-1} x}, \tag{3.115}$$

which is causal since $\boldsymbol{L_C}$ is lower triangular.

In the same way, starting from the UDL decomposition we can also generate a Cholesky factorization which results in the factorization of $\boldsymbol{R_x}$ into an upper triangular matrix followed by a lower triangular matrix,

$$\boldsymbol{R_x = U_C U_C^{*T}}, \tag{3.116}$$

where

$$\boldsymbol{U_C = (L_C)^{*T}}, \tag{3.117}$$

and which leads to the anti-causal transformation

$$\boldsymbol{y = U_C^{-1} x}. \tag{3.118}$$

Since both transformations at Equation (3.115) and (3.118) are special cases of the LDU factorization it can easily be shown that they also whiten a colored noise sequence.

4. QR Factorization

The Cholesky and LDU decompositions provide a means by which the normal equation can be solved by performing square root decompositions of the autocorrelation matrix $\boldsymbol{R}_x$. The QR factorization decomposes a data matrix $\boldsymbol{X}$ into an orthogonal matrix $\boldsymbol{Q}$ and an upper triangular matrix $\boldsymbol{R}$. Since this decomposition uses the data matrix directly, it provides better numerical accuracy than the square root methods which are based on the decomposition of the autocorrelation matrix $\boldsymbol{R}_x$ which is usually estimated from a finite length data [Ref. 17].

Let's consider the following derivation from [Ref. 14]. To begin, we consider the $K \times M$ data matrix

$$\boldsymbol{X} = \begin{bmatrix} | & | & \dots & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_M \\ | & | & \dots & | \end{bmatrix}, \tag{3.119}$$

where $\boldsymbol{x}_i = \left[x_i[1]\ x_i[2]\ \dots\ x_i[K]\right]^T$, for $i = 1 \dots M$, and the matrix is of full rank, meaning that the numbers of rows K is at least equal to the number of columns M and that these latter are independent. Since the columns are independent, the Gram-Schmidt orthogonalization procedure can be used to derive a set of M orthonormal vectors $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, $\boldsymbol{q}_3$,...,$\boldsymbol{q}_M$. For the Gram-Schmidt method, we choose a normalized version of the first column as the first orthonormal vector:

$$\boldsymbol{q}'_1 = \boldsymbol{x}_1 \tag{3.120}$$

and normalize it,

$$\boldsymbol{q}_1 = \frac{\boldsymbol{q}'_1}{\|\boldsymbol{q}'_1\|}. \tag{3.121}$$

To form the next orthonormal vector, we take the second column and remove from it the

component it contains which is in the q_1 direction, then normalize the result,

$$q'_2 = x_2 - (x_2^{*T} q_1) q_1, \tag{3.122}$$

and

$$q_2 = \frac{q'_2}{\|q'_2\|}. \tag{3.123}$$

For the l^{th} orthonormal vector, we have

$$q'_l = x_l - \sum_{i=l}^{l-1} (x_l^{*T} q_i) q_i, \tag{3.124}$$

and

$$q_l = \frac{q'_l}{\|q'_l\|}, \tag{3.125}$$

where $l \leq M$. Therefore, it can be seen that each q_l lies in the subspace defined by the respective x_l. In matrix format this can be shown as

$$\begin{bmatrix} | & | & \dots & | \\ | & | & \dots & | \\ q_1 & q_2 & \cdots & q_M \\ | & | & \dots & | \\ | & | & \dots & | \end{bmatrix} = \begin{bmatrix} | & | & \dots & | \\ | & | & \dots & | \\ x_1 & x_2 & \dots & x_M \\ | & | & \dots & | \\ | & | & \dots & | \end{bmatrix} \begin{bmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1M} \\ 0 & \rho_{22} & \cdots & \rho_{2M} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & \rho_{MM} \end{bmatrix}, \tag{3.126}$$

where ρ_{lm} represent the transformation coefficients that are applied to the data vectors to create the orthonormal vectors q_l. This equation can also be written as

$$Q = XR^{-1} \tag{3.127}$$

If we multiply both sides of Equation (3.127) with R,

$$\boldsymbol{X} = \boldsymbol{QR}. \tag{3.128}$$

This last equation now expresses the matrix $\boldsymbol{X}$ as a function of a rectangular matrix $\boldsymbol{Q}$ whose columns are orthonormal, and a matrix $\boldsymbol{R}$ which is square upper triangular.

To show that the QR decomposition can be used to obtain the triangular decomposition of an estimated correlation matrix defined by

$$\boldsymbol{R}_x = \frac{1}{N}\boldsymbol{X}^{*T}\boldsymbol{X}. \tag{3.129}$$

With (3.128) this becomes

$$\boldsymbol{R}_x = \frac{1}{N}\boldsymbol{R}^{*T}\boldsymbol{Q}^{*T}\boldsymbol{QR}. \tag{3.130}$$

Since $\boldsymbol{Q}$ is an orthonormal matrix,

$$\boldsymbol{Q}^{*T}\boldsymbol{Q} = \boldsymbol{I}. \tag{3.131}$$

Therefore Equation (3.130) becomes

$$\boldsymbol{R}_x = \frac{1}{N}\boldsymbol{R}^{*T}\boldsymbol{R}. \tag{3.132}$$

$\boldsymbol{R}_x$ can also be factored by LDU decomposition,

$$\boldsymbol{R}_x = \boldsymbol{L}\boldsymbol{D}_L\boldsymbol{L}^{*T}. \tag{3.133}$$

So it follows that

$$\frac{1}{N}\boldsymbol{R}^{*T}\boldsymbol{R} = \boldsymbol{L}\boldsymbol{D}_L\boldsymbol{L}^{*T}, \tag{3.134}$$

resulting in

$$\boldsymbol{R}^{*T} = \sqrt{N}\boldsymbol{L}(\boldsymbol{D}_L)^{1/2}, \tag{3.135}$$

and,

$$\boldsymbol{R} = \sqrt{N}(\boldsymbol{D}_L)^{1/2}\boldsymbol{L}^{*T}. \tag{3.136}$$

If this equation is solved for $\boldsymbol{D}_L^{1/2}$, we have

$$\boldsymbol{D}_L^{1/2} = \frac{1}{\sqrt{N}}(\boldsymbol{L}^{*T})^{-1}\boldsymbol{R}. \tag{3.137}$$

Because the matrix $\boldsymbol{L}^{*T}$ is unit upper triangular its inverse also has unit diagonal elements. Since the matrix $\boldsymbol{R}$ is upper triangular, the effect of performing the matrix multiplication $(\boldsymbol{L}^{*T})^{-1}\boldsymbol{R}$ results in a diagonal matrix which has the diagonal values of $\boldsymbol{R}$. This is denoted by $\boldsymbol{IR}$ which when put in (3.137) gives us

$$\boldsymbol{D}_L^{1/2} = \frac{1}{\sqrt{N}}\boldsymbol{IR}. \tag{3.138}$$

When we solve Equation (3.137) for $\boldsymbol{L}$, we get $\boldsymbol{L}$ expressed in terms of the QR factorization,

$$\boldsymbol{L} = \frac{1}{\sqrt{N}}\boldsymbol{R}^{*T}\boldsymbol{D}_L^{-1/2}, \tag{3.139}$$

where the formula for $\boldsymbol{D}_L^{1/2}$ is the previous one given. Then we can do the same transformation as for LDU,

$$\boldsymbol{y} = \boldsymbol{L}^{-1}\boldsymbol{x}. \tag{3.140}$$

Since the QR factorization directly uses the data matrix to compute the triangular matrix $\boldsymbol{L}$, it is not subject to the problems resulting from using an estimated correlation matrix. This makes this technique less subject to errors resulting from poor conditioning and usually leads to better numerical accuracy.

G. DIFFERENTIAL FILTER

In Chapter II, it is shown that the absorption of sound energy by sea water has an important effect on the ambient noise in the ocean. The intensity attenuation coefficient is defined as being proportional to the square of the frequency for most frequencies. This resulted in a sound intensity magnitude proportional to frequency squared,

$$I \propto \frac{1}{f^2}, \tag{3.141}$$

and the pressure magnitude,

$$P \propto \frac{1}{f}. \tag{3.142}$$

For the noise to be white, it is required that the magnitude of the frequency response be equal to a constant. The colored noise signal $n_c[n]$ is related to its frequency response $S_{n_c}[\omega]$ by the Fourier transform

$$F\{n_c[n]\} = S_{n_c}[\omega]. \tag{3.143}$$

The frequency response is equal to

$$S_{n_c} = A\frac{2\pi}{\omega}e^{j\omega t}, \tag{3.144}$$

where A is a constant. If the random process $n_c[n]$ is differentiated with respect to time, in the frequency domain this has the effect of multiplying $S_{n_c}[\omega]$ by a factor of $j\omega$;

$$F\left\{\frac{d}{dt}n_c[n]\right\} = \frac{d}{dt}S_{n_c}[\omega] \tag{3.145}$$

$$= \frac{d}{dt}A\frac{2\pi}{\omega}e^{j\omega t} \tag{3.146}$$

$$= A\frac{2\pi}{\omega}\frac{d}{dt}(e^{j\omega t}), \tag{3.147}$$

where $\omega = 2\pi f$, resulting in

$$F\left\{\frac{d}{dt}n_c[n]\right\} = j2\pi A e^{j\omega t}. \tag{3.148}$$

This last equation shows that the frequency spectrum of the time differentiation of $n_c[n]$ should have a constant magnitude equal to $2\pi A$ and therefore has been whitened. Since the differential operator is a linear operation, at the frequency of a given signal, the

ratio between signal power and noise power should remain the same and the noise is whitened.

H. WHITENING PROPERTY OF PREDICTION-ERROR FILTER

A prediction error filter (PEF) is shown in Figure 3.8.

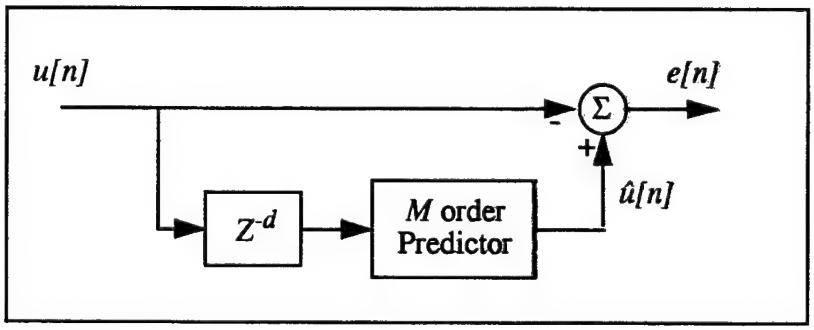

Figure 3.8: Prediction-error filter

The PEF combines the features of a predictor filter whose role it is to estimate the value of the sample $u[n]$ based on the M previous values starting at a delay of d. The error between the predicted value and the real value is then calculated by the summer. Often a variation of this set-up is used which computes the M weights of the predictor in an adaptive fashion. The advantage to the adaptive filter is as its name implies is that it has the ability to change its behavior as the environment changes. For example in the SONAR processing problem, the characteristics of the ambient noise may change with variations in locations, weather conditions, time of day, and, movements of noise generators. The adaptive filter will change its weight to ensure that its error is minimized. In those cases, Figure 3.8 is modified as shown on Figure 3.9.

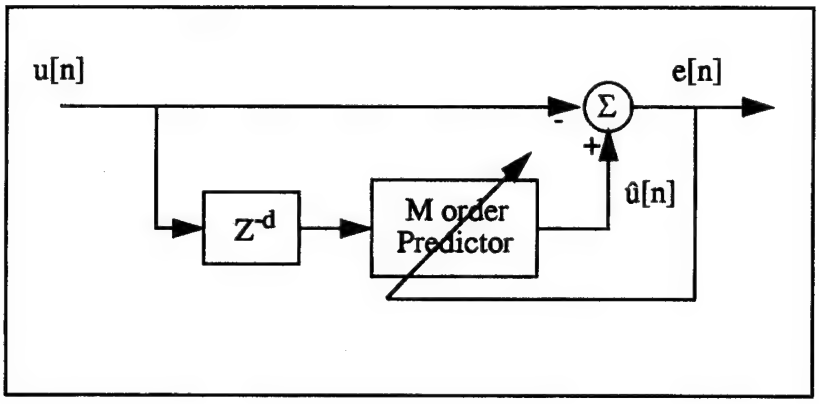

Figure 3.9: Adaptive PEF

As shown, the errors are used to update the weights of the predictor to minimize the difference between the real signal and its estimate. This type of filter relies on the correlation characteristics of the components that are being separated. In our case, the two components are the noise and the known signal that is to be detected.

In the case of white noise, noise samples with a delay of one or more between them are uncorrelated. Therefore, the predictor can be used to effectively estimate the values of the known signal if the delay is chosen at a point where the correlation function of the signal is at a maximum.

In the case of colored noise, the noise and the signal may have correlation function which are less distinctive and additional care must be taken in the design of the filter. The correlation function of the low-pass Butterworth noise for the first 25 delays is shown in Figure 3.10.

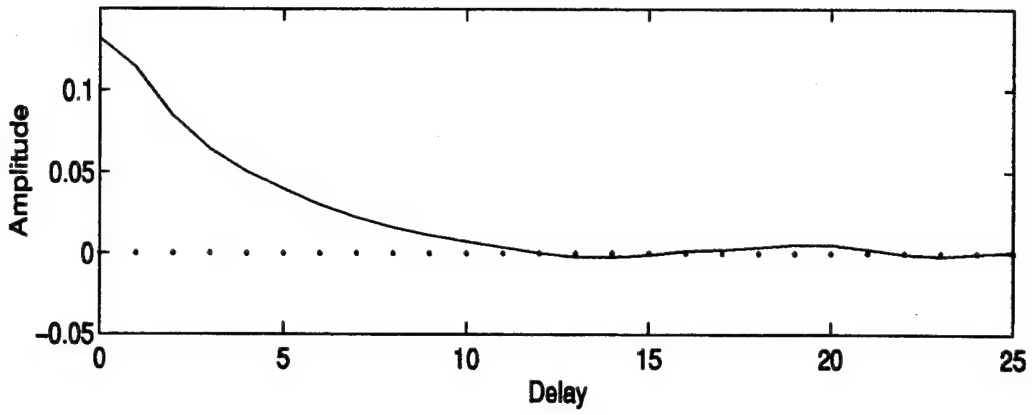

Figure 3.10: Correlation function of experimental first order low-pass Butterworth colored noise.

As can be readily observed, the correlation function for colored noise is a relatively smooth function which does not oscillate around zero at predetermined delays. Different realizations of the noise result in different zero crossovers but the general form of the function is maintained and relatively strong correlation is present only at short delays. However, because the known signal to be detected is a periodic signal, its correlation function is also periodic and as such goes through zero at known delays. This may allow a delay to be chosen that decorrelates the signal while maximizing the correlation of the noise. If the signal is decorrelated, the predictor can provide a reasonable estimate of the colored noise. In this case, the error between the estimated noise value and the real value consists of the signal and a prediction error. The prediction error is based on two factors; the amount of decorrelation between the noise samples at the chosen delay, and, a quantization error. The predictor relies on the correlation between adjacent samples of the input process. Therefore, when the order of the filter is increased, the correlation of the adjacent samples of the input process is successively reduced until the filter is of an order where the output samples are uncorrelated and therefore white [Ref. 18: p. 216]. However, so that the desired signal is not whitened, the PEF requires that the delay selected be such that the correlation of the component to be predicted, in our case the colored noise, be as

large as possible while that of the signal has a correlation as close to zero as possible. Any amount of correlation at that delay in the desired signal causes the predictor weights to converge to a state where some energy of the other signal is present in the estimated quantity and is therefore removed from the "error" signal. Therefore, in our problem, the predictor estimates the noise component of the received signal. If the desired signal is uncorrelated at the delay chosen, then the error $e[n]$ consists in the desired signal and a white noise prediction error. The filter whitens the noise corrupting the signal if the error is chosen as the desired output. The variance of the error depends on the magnitude of the learning rate parameter. To a certain extent, the smaller the parameter is, the lower is the variance of the prediction error but the filter weights of the predictor take longer to converge and are less responsive to sudden variations in the acoustic environment. Conversely, if the learning rate parameter is larger, the weights converge more quickly but the prediction errors are larger resulting in white noise of higher variance in the predicted error and therefore a lower output signal to noise ratio.

IV. COLORED NOISE DETECTION RESULTS

This chapter presents the results achieved with the various methods addressed in Chapter III. Each whitening technique is addressed in the order in which it is discussed in Chapter III beginning with the matched filter and inverse filter, and closing with the prewhiteners.

A. TEST PROCEDURES

A common test sequence is used for the testing of all methods except the prediction-error filter which uses a different signal. This sequence consists of three sinusoids of different frequencies corrupted by colored noise. The sinusoids form the known signal that is to be detected. They are located in different areas of the spectrum to allow a simultaneous view of the effects at each spectral location. One sinusoid is at a very low frequency where the slope of the noise is greater, one is at higher frequencies where the noise spectral density is almost flat, and one is located at an intermediate frequency. The frequencies of the sinusoids are $0.05\,fs$, $0.15\,fs$, and, $0.3\,fs$.

The noise is created by passing a white Gaussian noise through a low-pass filter. It is added to the signal of interest and is colored with a first order low-pass Butterworth filter with a cut-off frequency of $0.1\,fs$. The Butterworth filter is an all-pole filter. Its magnitude-square frequency response is given by [Ref. 19]

$$|H(\omega)|^2 = \frac{1}{1+(\omega/\omega_c)^{2N}}, \tag{4.1}$$

where N is the filter order, ω the digital frequency and ω_c the 3dB cut-off digital frequency. Therefore, a first order Butterworth filter has a frequency response proportional to

$$|H(\omega)|^2 \propto \frac{1}{\omega^2}, \tag{4.2}$$

which is similar to the frequency response observed for the ambient noise in the ocean. Since a Butterworth filter is used to create the colored noise, the theoretical power spectral

density and correlation function of the noise is known. The power in the signal is chosen so that the spectral heights of each of the sinusoids and the noise have a ratio of approximately two at the respective spectral locations, when the white Gaussian noise used as a input to the Butterworth noise filter has a variance of 100. Therefore, the three sinusoids forming the signal have amplitudes of 0.475, 0.2, and, 0.07 respectively. Different methods require that higher input signal to noise ratios be used. For these results, the white noise power, i.e., its variance, used to create the colored noise, is reduced. This, in effect, reduces the power of the colored noise proportionally across the frequency spectrum.

Figure 4.1 shows an averaged (100 realizations) power spectral density of the test sequence with the variance of the white Gaussian noise set at 100.

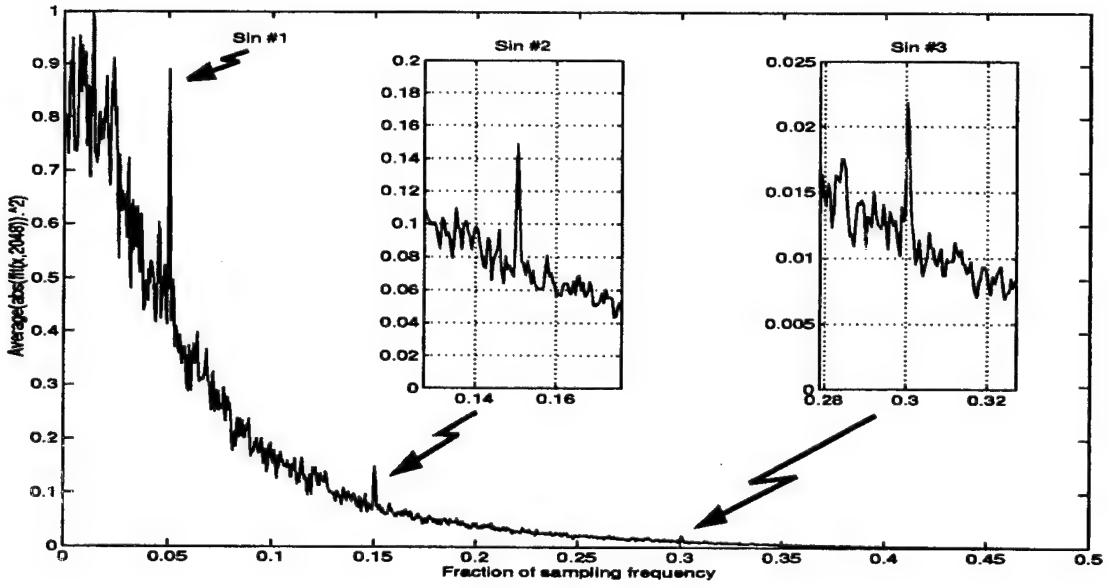

Figure 4.1: Normalized power spectral density of signal plus noise

The power spectral density in the area of Sinusoids #2 and #3 are also plotted separately on each applicable figure to better illustrate the relationship between the signal power and the noise power at those frequencies. This is also done in all result plots where relative power levels are very different from one part of the frequency spectrum to the

other. All plots are normalized to the largest peak since our interest is in the signal to noise ratios resulting from the transformations.

The power spectral density from each processing scheme is shown using averaged periodograms. The periodogram is a well known estimate for the power spectral density. Ten realizations of each method are performed and the ten results averaged. This is done because the periodogram is by definition an unreliable estimator due to its large variance. When the results of ten periodograms are averaged, the variance is reduced by a factor of ten, rendering a more consistent estimate. In all cases the estimated power spectral density of the sequence before and after whitening are shown. If the transformation effectively decorrelates the noise components of the samples, the three sinusoids after decorrelation should all have approximately the same signal to noise ratio at their respective frequency.

In all cases the processing consists of the following steps:

Step a: Add the signal and colored noise to create the test sequence.

Step b: Derive the transformation vector or matrix depending on the algorithm being tested.

Step c: Operate on the test sequence the applicable transformation, and create a new sequence in which the noise has been whitened.

Step d. Use the fast-Fourier transform operation to form a bank of discrete detectors to attempt to locate the three signal components in the white, or whitened noise. The results are computed as periodograms.

Step e: Run the transformation of ten different realizations of the experiment and compute an averaged periodogram.

In terms of detection, the resulting power spectral density, for two signals in noise, s_1 and s_2 in this example, is interpreted as shown on Figure 4.2:

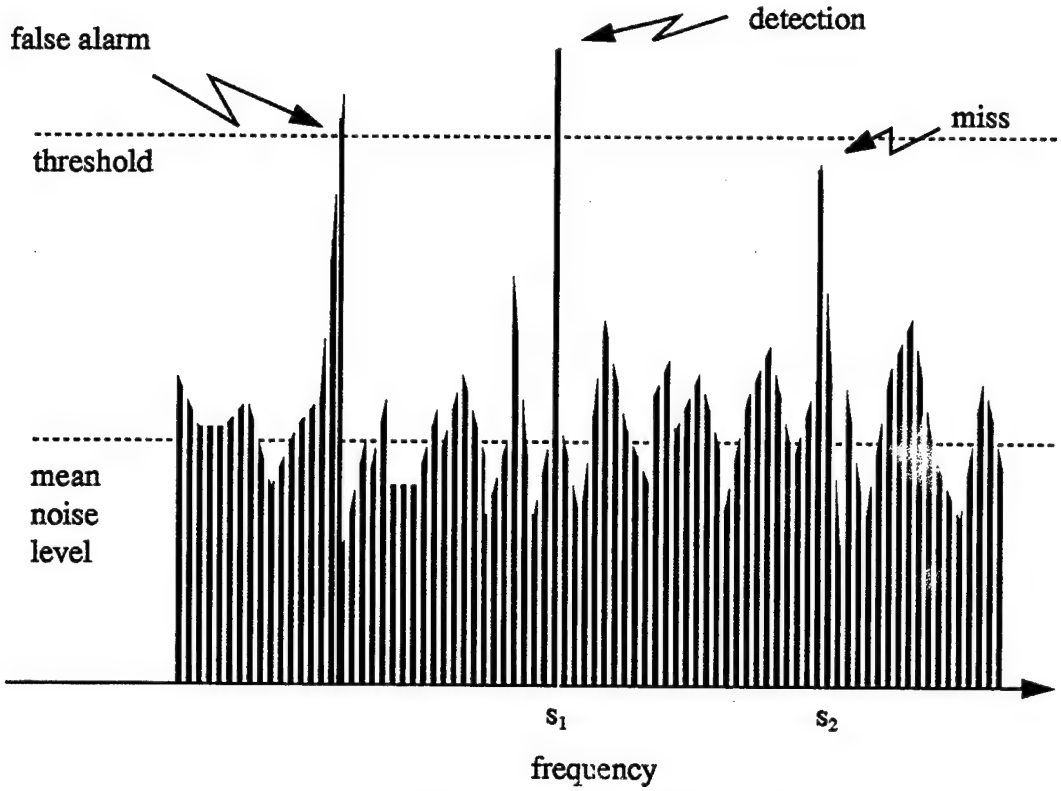

Figure 4.2: Detection via power spectral density

A detection threshold will not be set per se but rather we will be looking at how the power ratio between the signal and the noise at the respective frequencies of the three sinusoids are maintained by each transformation.

1. Estimates of the Correlation Matrix

Many of the methods that are discussed require that the correlation function of the noise be known or estimated. We assume that the power spectral density of the ambient noise is known.

When the power spectral density of the noise is known, its autocorrelation function can be estimated using the Wiener-Khintchine relation [Ref. 19].

Let $S_x(e^{j\omega})$ be the power spectral density of the wide sense stationary random vector x. The autocorrelation is related to the power spectral density as follow:

$$S_x(e^{j\omega}) = \sum_{l=-\infty}^{\infty} R_x[l]e^{-j\omega l} \tag{4.3}$$

which is the discrete time Fourier transform of the autocorrelation function. Since S_x and R_x are DTFT pairs, the latter can be given by:

$$R_x[l] = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_x(e^{j\omega})e^{j\omega l}d\omega \tag{4.4}$$

It is reasonable to assume that the noise power spectral density in a given area of ocean is known with a certain degree of accuracy as a result of previous data and knowledge of the present environmental conditions. In most of the tests, it is assumed that the power spectral density of the noise is known. From this power spectral density, the corresponding autocorrelation function can be derived using the Wiener-Khintchine relationship.

2. Matrix Decomposition Methods

These methods are realized by using block transformations to filter the sequence. This means that the resultant sequence is formed by one matrix multiplication obtained via the product of a matrix of orthonormal vectors and the input sequence. One of the problems with using this technique is that it is usually preferable to use larger matrices allowing more input data. This implies that the correlation matrix to be decomposed is also larger. The processing resources required to factor larger matrices into its required orthonormal constituents becomes quickly prohibitive. To attempt to mitigate this problem, the correlation matrices used in these methods are limited to a size of 256 X 256. This means that the multiplications are also performed in blocks of 256 points. Since we want to be able to compare the results achieved with all methods, and we want representative results, all power spectral densities shown are for data lengths of 1024. For matrices factorization methods limited by size, four transformed block of data are then joined to form one 1024

length sequence. It is expected that the results may be distorted since the transformed data points starting at n, n+256, n+512, and, n+768, would then all be the results of the same orthonormal vector multiplying different input sequences. However, when only four blocks are used, this problem is not observed. When smaller matrices and therefore more data blocks are used in an attempt to reduce the processing requirements, a clear periodic pattern emerged based on the number of blocks used. Therefore, the use of many small block transforms leads to unsatisfactory results and is avoided.

B. RESULTS OF OPTIMAL DETECTION

1. Discrete Time Matched Filter for Colored Noise

This section addresses the results for the discrete time matched filter for colored noise as derived in Chapter III. The figures shown were computed using the Matlab program given in Appendix H. The results are achieved by filtering the test sequence $\boldsymbol{x}$ with $\boldsymbol{h}$ to arrive at a sequence $\boldsymbol{y}$. The transfer function $\boldsymbol{h}$ of the filter is computed using Equation 3.56.

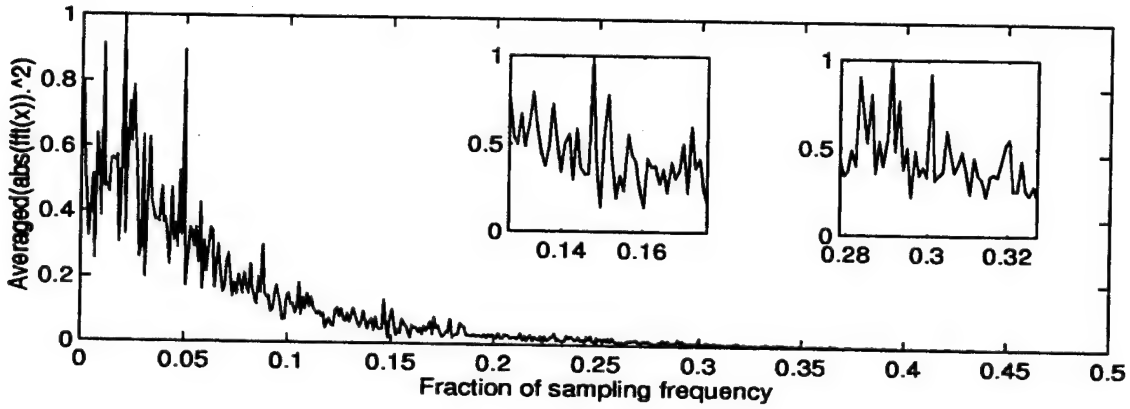

Figure 4.3: Normalized power spectral density of test sequence (white Gaussian noise variance=100).

Figure 4.3 shows the normalized averaged power spectral density for the input sequence $\boldsymbol{x}$. As stated previously, the three sinusoids are at frequencies 0.05 fs, 0.15 fs, and, 0.3 fs. The variance of the white Gaussian noise used to create the colored noise is 100. The power in each of the three sinusoids is selected so that the spectral heights of signal and

noise at each frequency have a ratio of approximately two. Figure 4.3 shows an averaged periodogram based on ten realizations rater than 100 realizations as in Figure 4.1. The variance of the periodogram is evident when comparing Figures 4.1 and 4.3. In Figure 4.3, Sinusoid #2 and Sinusoid #3 are not distinguishable from the noise background. Sinusoid #1 can be identified but other peaks at lower frequencies are clearly stronger which makes its detection unlikely with the periodogram.

The matched filter is usually used to generate one number whose value is compared to a threshold to determine the presence or absence of a signal. Because of the equivalence of the matched filter and the correlator, the output of the matched filter can be interpreted as a correlator. Therefore, because we want to compare all results in terms of averaged periodograms, the results of the matched filter detection are shown in the form of a correlator. The averaged periodogram is computed for the sequence consisting of all matched filter results. The next figure shows the normalized averaged power spectral density of the filtered sequence y.

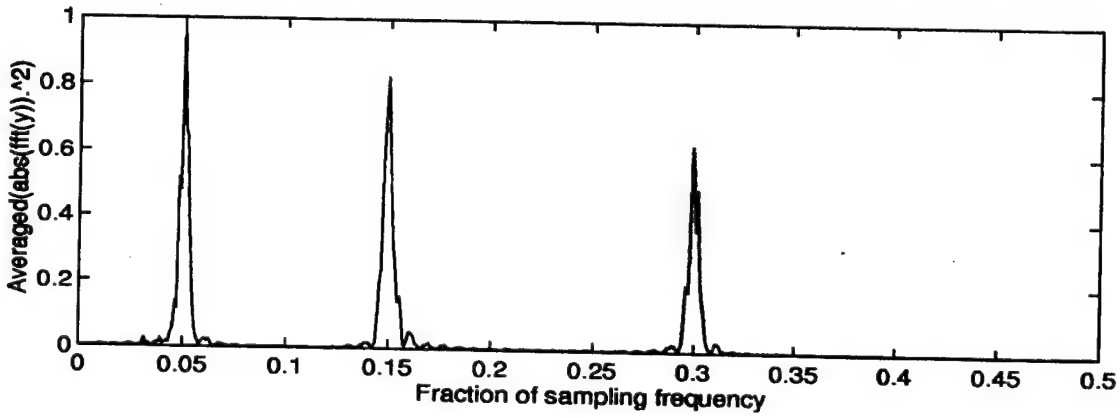

Figure 4.4: Normalized power spectral density of output of matched filter.

Figure 4.4 clearly shows the success of the algorithm in filtering the colored noise. The strong relative power of each sinusoid to the noise is well shown. All three sinusoids have significant signal to noise ratios, far better than those in Figure 4.3. The peaks are sharp and precisely located at the correct frequencies. The effect of the colored noise appears to have

been almost completely eliminated with the exception of some residual noise around each sinusoid. A detection threshold could be established at a very low level with no chances of false alarms or of misses. These results show that the matched filter for colored noise applications is very successful.

2. Inverse Filter

The results achieved for the inverse filter derived in Chapter III are shown here. These results were achieved using the Matlab program shown in Appendix I. The inverse filter is used as a prewhitener. The results where achieved by running the test sequence $\boldsymbol{x}$ through the filter point by point to arrive at a sequence $\boldsymbol{y}$ consisting in mostly signal. The transfer function $\boldsymbol{h}$ of the filter is computed with Equation 3.66 by assuming that the power spectral density of the colored noise and therefore the transfer function responsible for the colorations of the noise is known.

Figure 4.5 shows the normalized averaged power spectral density for the input sequence. In this case, the noise power is reduced by setting the white Gaussian noise variance to 25. Compared to the case when the white Gaussian noise variance is set to 100, the use of the lower variance provides a gain of approximately 6 dB. This gain is shown clearly in Figure 4.5 where the signal peaks are much more clearly distinguishable than in Figures 4.1 and 4.3. In fact Sinusoid #1 could be reliably detected with a simple detection threshold despite the higher noise levels at lower frequencies. Sinusoid #2 and #3 however have far less power than the low frequency noise and Sinusoid #1.

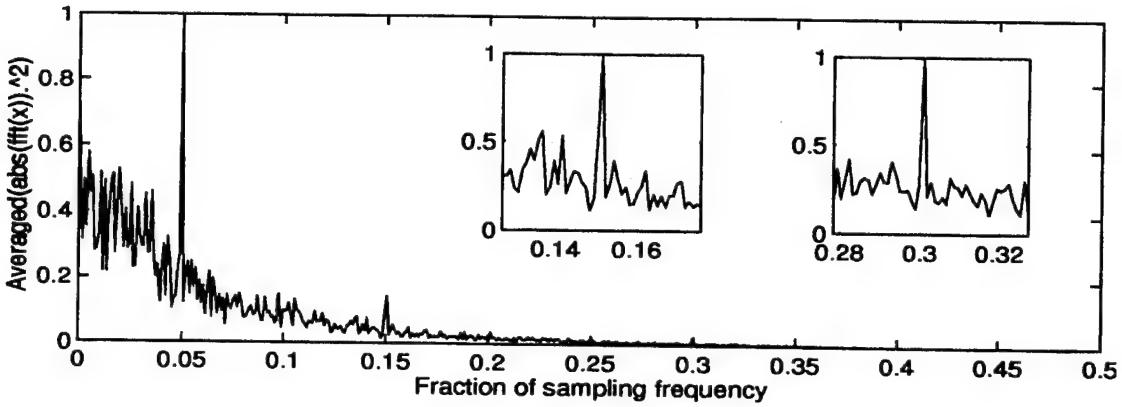

Figure 4.5: power spectral density of signal + colored noise (white Gaussian noise variance=25)

The results of filtering the sequence x with the inverse filter are shown at Figure 4.6:

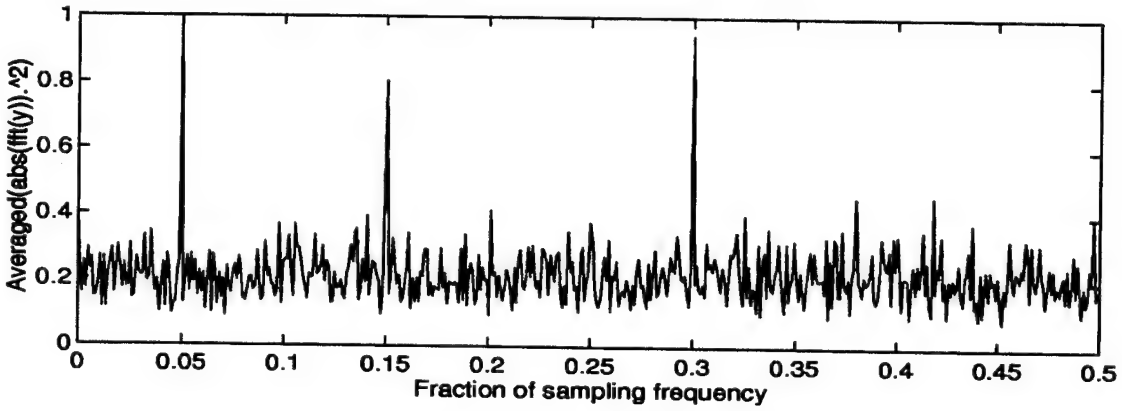

Figure 4.6: Normalized power spectral density of whitened signal using inverse filter.

We observe that the noise has been completely whitened as theoretically predicted. It can also be seen that the three components of the signal can be clearly detected and have approximately the same relative SNR at their respective frequency locations. Because the inverse filter is essentially an equalizer, the lower power sinusoids have been "raised" to approximately the same level as the first. Commensurately, the noise at low frequencies is reduced and that at higher frequency is increased. The transformation does not appear to produce peaks at incorrect frequencies which could lead to false alarms. Since the inverse

filter does not require decomposition of large matrices or matrix multiplications, the processing requirements for this whitener are relatively minimal.

3. Eigenvector Factorization

The eigenvector factorization results are the first of the matrix decomposition methods that are shown. The results achieved for the eigenvector transformation derived in Chapter III are shown here. These results were obtained using the Matlab program shown in Appendix J. The results are achieved by separating the test sequence x into four blocks and multiplying each of these by the orthonormal matrix of eigenvectors of the theoretical noise correlation matrix of size 256 X 256.

Figure 4.7 shows the normalized power spectral density for the input sequence. The noise power is reduced by setting the white Gaussian noise variance to 10. This is done due to the poor performance of the whitener as is shown on Figure 4.8. The low noise power is clearly apparent in Figure 4.7 by the clear unambiguous signal peaks and the relative lack of significant noise levels. In fact Sinusoid #1 can be reliably detected with a simple detection threshold despite the higher noise levels at lower frequencies. We note that Sinusoid #2 and Sinusoid #3 would be also be easily detected by common bandpass filters with predetermined detection thresholds.

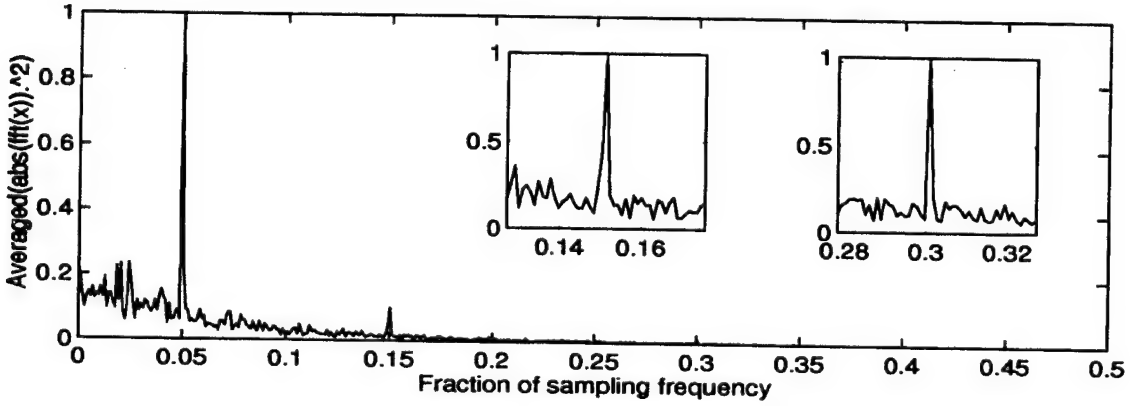

Figure 4.7: Normalized power spectral density of signal plus colored noise (white Gaussian noise variance=10).

The result achieved with the noise matrix eigenvector transformation is shown in Figure 4.8:

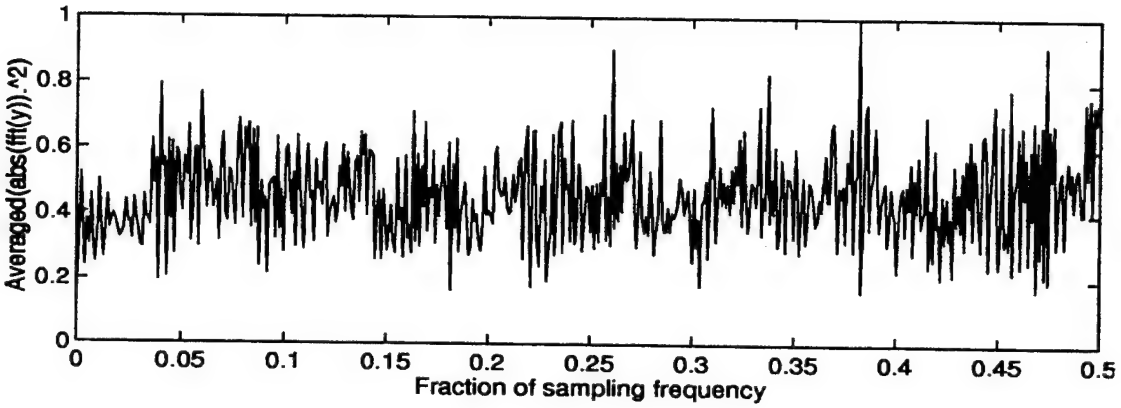

Figure 4.8: Normalized power spectral density of whitened signal using eigendecomposition.

Figure 4.8 shows that the transformation has whitened both the signal and the noise components. None of the signal remains despite low noise power, and detection would not be possible. The reason for this is due to the singular condition of the noise autocorrelation matrix. Although the rank of this matrix equals 256, its determinant equals zero. This result is unexpected and may be a function of the magnitudes of the values in the correlation matrix. The decomposition of the matrix results in eigenvalues which are less than one and where many are on the order of 10^{-8}. The eigenvector decomposition method is particularly susceptible to this problem. The singular value decomposition technique discussed next provides a way to accurately compute the eigenvectors of a singular matrix.

4. Singular Value Decomposition (SVD)

As stated earlier, the SVD technique provides an alternative way to compute the matrix of eigenvectors for a given correlation matrix without having to compute it first. It does so by using a matrix of data. In this experiment the data used is that of the colored noise. It is assumed that sequences of recorded noise have been obtained for the area in question and that this noise is representative of the actual conditions. The noise data matrix

used is of dimension 48 X 1024, and is created by using noise data of different realizations than that used in the input sequence x. From this noise data matrix, the SVD algorithm computes a matrix of eigenvectors and a matrix of singular values. The matrix of eigenvectors is then used in the same way as for the previous experiment to decorrelate the noise component of the received sequence. The figures shown were computed using the Matlab program given in Appendix K.

Figure 4.9 shows the normalized averaged power spectral density for the input sequence whitened via the SVD decomposition. In this case the noise power is set by using a white Gaussian noise variance of 50.

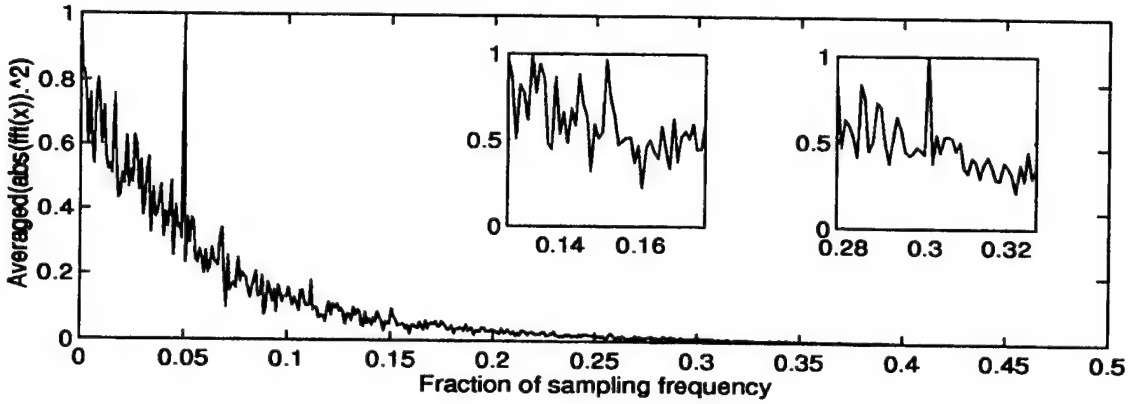

Figure 4.9: Normalized power spectral density of signal+colored noise (white Gaussian noise variance=50).

As can be observed, the noise power is such that the sinusoids are even difficult to distinguish with the averaged periodogram. Even the use of a bandpass filters around the frequencies of interest or a low pass filter removing the stronger noise at lower frequencies would make detection very difficult.

Figure 4.10 shows the averaged power spectral density of the transformed sequence.

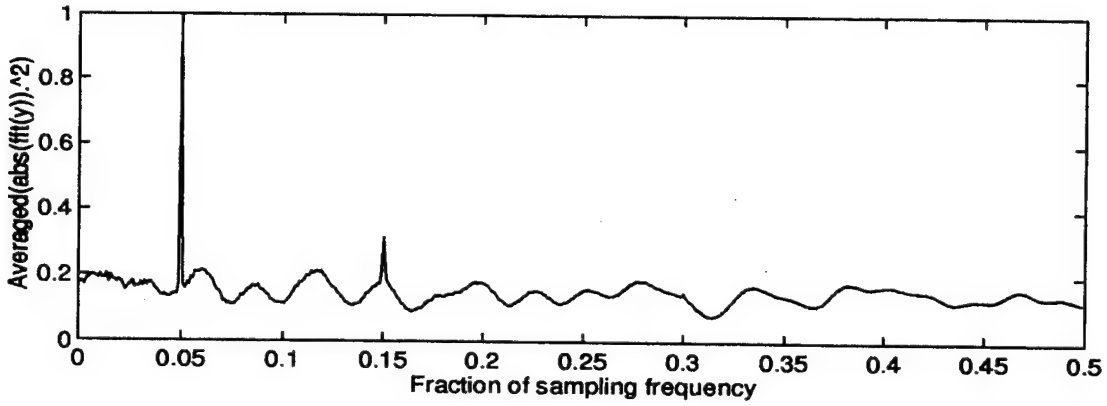

Figure 4.10: Normalized power spectral density of whitened signal with SVD.

It can be observed that the results are far better with the SVD method than with the eigenvector factorization method despite the fact that both methods, in theory, should arrive at the matrix of eigenvector of the same correlation matrix. Sinusoid #1 and Sinusoid #2 are readily detectable. Sinusoid #3 however only appears as a small protrusion and would not be detectable. This result demonstrates the superiority of the SVD algorithm when the correlation matrix of the noise is poorly conditioned or singular. As stated in Chapter III, the SVD method is far more robust than the normal eigenvector/eigenvalue decomposition method and does provide the means to achieve optimal detection as a prewhitener.

5. Mahalanobis Transformation

The Mahalanobis transformation is another transformation method based on the eigendecomposition of the theoretical noise correlation matrix. The results achieved for this transformation are shown here. These results were computed using the Matlab program shown in Appendix L. The results where achieved by separating the test sequence $\boldsymbol{x}$ in four blocks and transforming each of these according to Equation 3.96.

Figure 4.11 shows the normalized power spectral density for the input sequence. The noise power is reduced by setting the white Gaussian noise variance to 25.

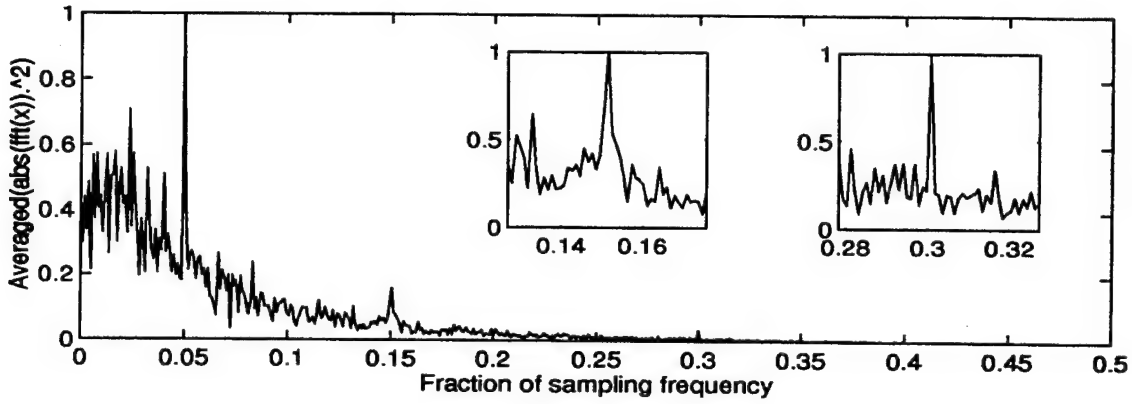

Figure 4.11: Normalized power spectral density of test sequence (white Gaussian noise variance=25).

Figure 4.12 shows the power spectral density of the transformed sequence.

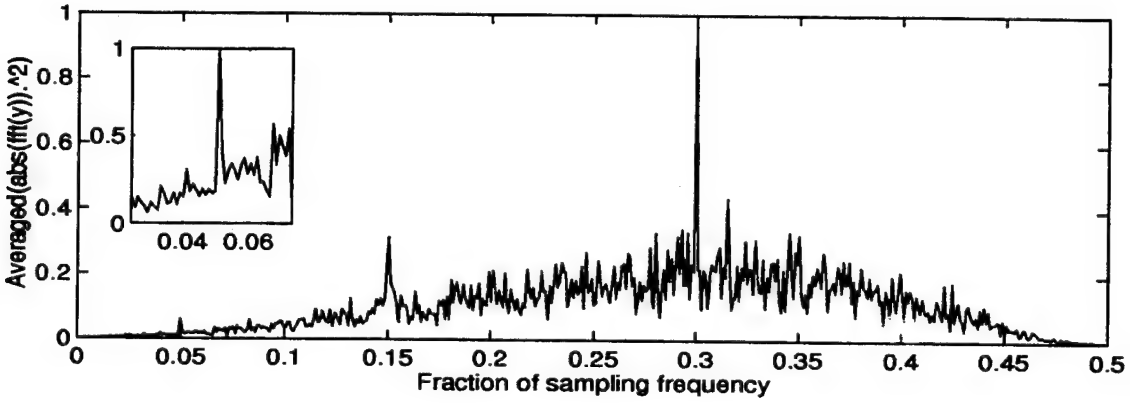

Figure 4.12: Normalized power spectral density of whitened signal using Mahalanobis transformation.

It can be observed that although the noise has maintained some correlation, it has been considerably whitened when compared to Figure 4.11. At the higher frequencies where Sinusoid #3 is located, the transformation has maintained its local signal to noise ratio in a way in which it would be easily detected. At lower frequencies where Sinusoid #1 and #2 are, the signal to noise ratio at that location along the frequency axis appears to have been maintained but since the noise floor continues rising until about 0.32 *fs*, and is more than

twice the height at that frequency than it is at 0.15 f_s, detection would require additional processing.

6. Lower-Upper Factorization (LDU)

The LDU factorization is the first of the whitening transformations to use a triangular matrix, that is shown here. Figure 4.13 shows the normalized power spectral density for the input sequence. In this case the noise power is achieved by using a white Gaussian noise variance of 25. The figures shown were computed using the Matlab program given in Appendix M.

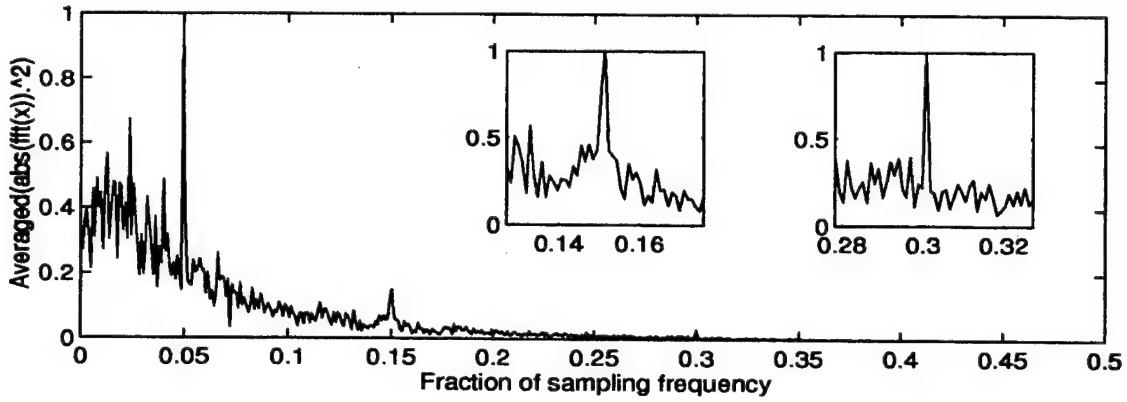

Figure 4.13: Normalized power spectral density of test sequence (white Gaussian noise variance=25).

This sequence is then whitened with the $\boldsymbol{L}$ triangular matrix of the LDU decomposition of the theoretical correlation matrix, in accordance with Equation 3.102.

Figure 4.14 presents the averaged power spectral density of the whitened sequence.

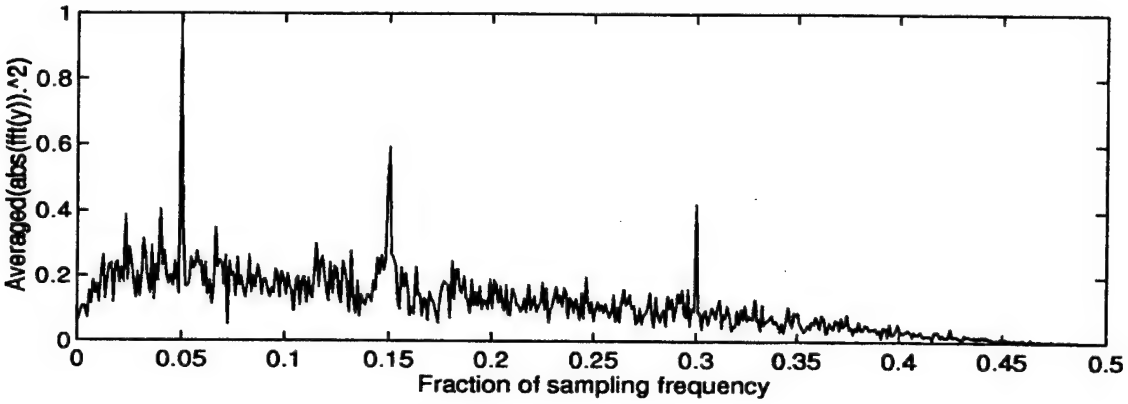

Figure 4.14: Normalized power spectral density of whitened sequence using LDU.

It can be observed that the transformation has not succeeded in completely whitening the input sequence. From 0 to 0.02 f_s there is an upwards curve followed by a steady descending slope from 0.025 to 0.5 f_s. However, the three sinusoids are clearly distinguishable with few occasional peaks which could lead to false alarms. Overall the results show that the local signal to noise ratio has been maintained or somewhat improved but that the noise is not fully whitened especially at very low frequencies where the slope of the noise is greatest.

7. Upper-Lower Factorization (UDL)

This transformation is essentially the same as above but it uses a upper triangular matrix. This leads to the anti-causal transformation of the theoretical noise correlation matrix shown at Equation 3.111. The results of this whitening on the same input sequence used for the LDU factorization, shown at Figure 4.13, is shown at Figure 4.15.

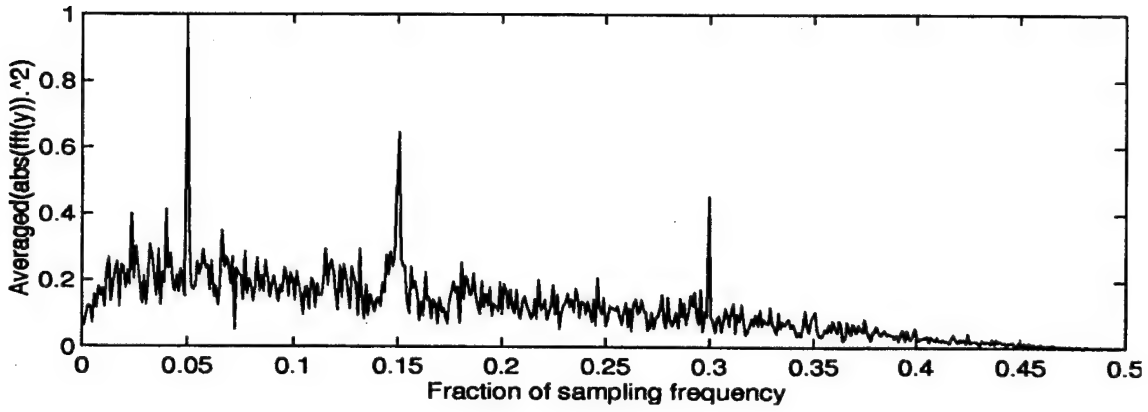

Figure 4.15: Normalized power spectral density of whitened sequence using UDL.

It can be observed that the results achieved with the UDL transformation is very similar to that achieved with the LDU transformation. The anti-causality of the transformation seems to have no effect on the performance of the whitener.

8. Cholesky Factorization

As stated in Chapter III, the Cholesky decomposition is based on the LDU and UDL decompositions. As such, results achieved with it are expected to be similar to those resulting from those methods. As is done for the LDU and UDL decompositions, a white Gaussian noise variance of 25 is used to create the colored noise and results in the power spectral density for the input sequence shown at Figure 4.16.

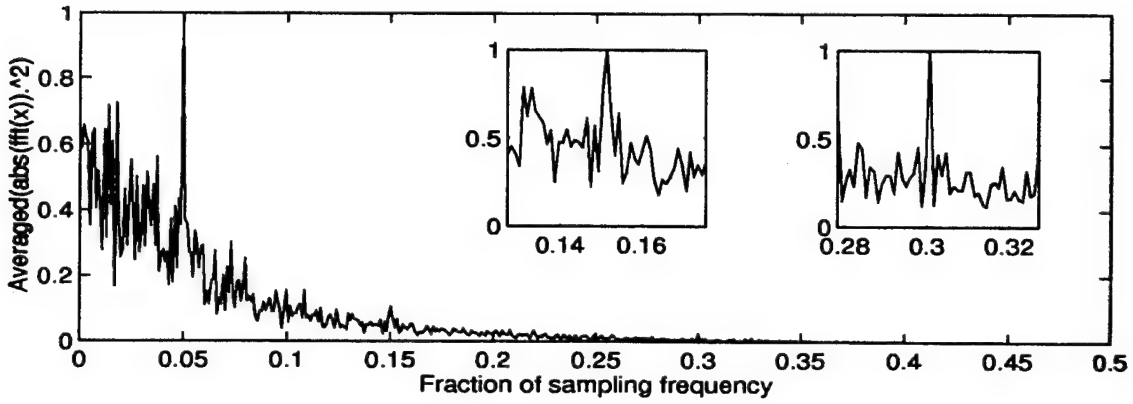

Figure 4.16: Normalized power spectral density of test sequence (white Gaussian noise variance=25).

The next two figures present in turn the causal and anti-causal Cholesky transformation results, related respectively to the LDU and UDL factorizations. Figure 4.17, shows the power spectral density of the transformed sequence using the Cholesky factor consisting of a lower triangular matrix, therefore resulting in a causal transformation. This transformation is performed using Equation 3.115. The Matlab program used to perform the transformations and generate the figures is included at Appendix N.

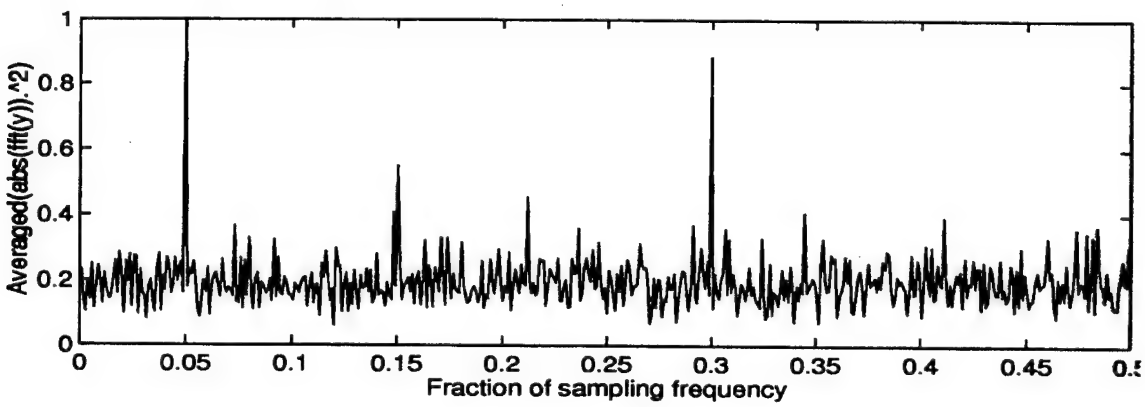

Figure 4.17: Normalized power spectral density of whitened sequence using lower triangular Cholesky factor.

Figure 4.18 presents the averaged power spectral density of the transformed sequence using the upper triangular Cholesky matrix. This transformation is anti-causal.

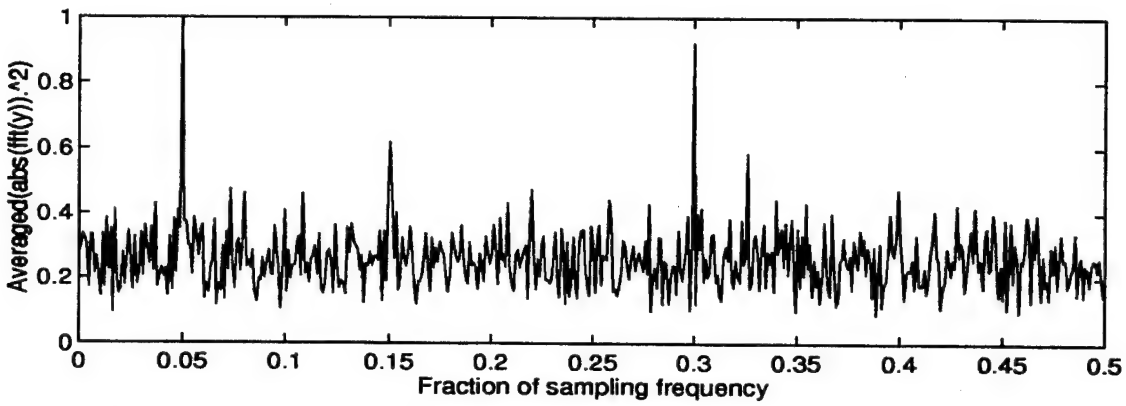

Figure 4.18: Normalized power spectral density of whitened sequence using upper Cholesky factor.

We can see that the results achieved are quite good in that the noise is whitened and the ratio of the spectral heights of each of the sinusoids and the noise at the respective spectral locations are also maintained. Some spurious peaks may cause false alarms for Sinusoid #2. The causality or anti-causality of the transformation seems to have no effect on the performance achieved. The three sinusoids are reliably detected and have maintained or improved their local SNR relative to the input sequence. Since the Cholesky decomposition differs from the LDU decomposition in that the square root of the diagonal matrix $\boldsymbol{D_L}$ scales the matrix $\boldsymbol{L}$, this last factor appears to allow the method to correctly whiten the noise.

9. QR Factorization

QR factorization, like SVD makes use of a matrix of data to arrive at the lower triangular matrix $\boldsymbol{L}$ of the LDU factorization. In this case, a square noise only data matrix of dimensions 256 X 256 is used. In the same way as for the SVD method, the noise used in the data matrix is independent of that used in the sequence $\boldsymbol{x}$. Four block transforms of length 256 are joined to form the whitened sequence of 1024 points. The Matlab program used to generate the results is shown at Appendix O.

Figure 4.19 shows the power spectral density for the input sequence. The white Gaussian noise variance is set to 50.

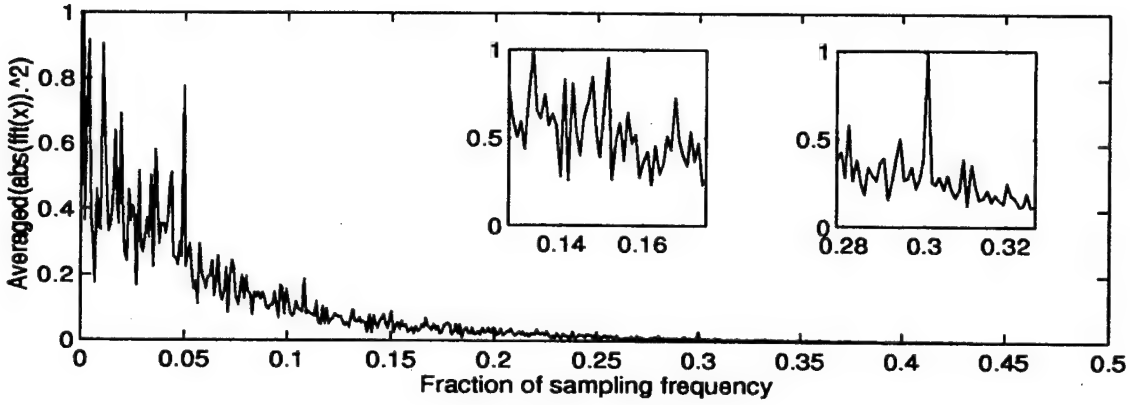

Figure 4.19: Normalized power spectral density of test sequence (white Gaussian noise variance=50.

One point to note concerning Figure 4.19 is that Sinusoid #2 is not well resolved in this realization of the process. This result is also observed in the whitened output shown in Figure 4.20.

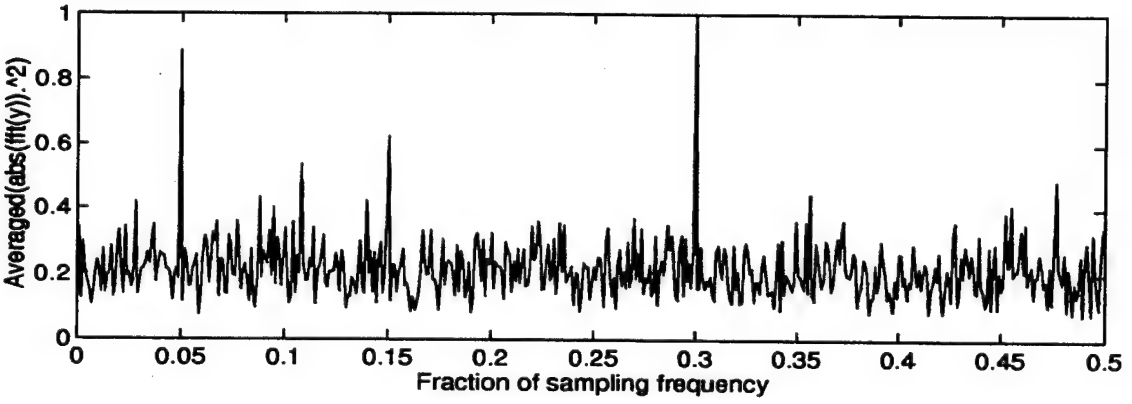

Figure 4.20: Normalized power spectral density of whitened signal using the QR factorization to achieve an estimate of the L matrice. (256X256)

Figure 4.20 shows that the colored noise is successfully whitened. All three sinusoids are identifiable although the variance of the estimate may still lead to some false alarms depending on the height of the detection threshold. As stated above, the Sinusoid #2 peak

is not as detectable as Sinusoid #1 and Sinusoid #3 but this is also the case with the input sequence. The local signal to noise ratio of all three sinusoids has also been improved compared to the averaged periodogram shown in Figure 4.19. This result seems to indicate that the numerical accuracy required to perform successful whitening with the transformations discussed herein is important and is not achievable from the use of the theoretical noise correlation matrix. Overall this is the most successful whitener among those using matrix decomposition methods in terms of truly whitening the colored noise.

C. RESULTS OF OTHER NON-OPTIMUM FILTERS

1. Differential Filter

This section shows the results of using the simple differentiation function to attempt whitening as discussed in Chapter III. The advantage in using this method, should it be successful, lies in its simplicity and efficiency. It does not require any matrix factorization process nor does it demand numerous multiplications, but it can only be used in certain colored noise situations (i.e., $1/f$, $1/f^2$ noise, etc.). The figures shown were computed using the Matlab program given in Appendix H.

The averaged periodogram for the input sequence of signal plus noise is shown in Figure 4.21. For this experiment the white Gaussian noise variance is set to 25.

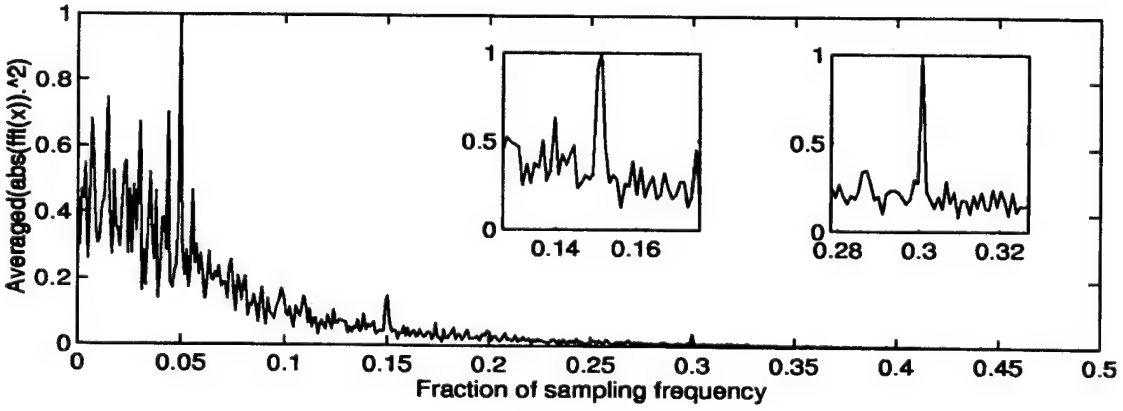

Figure 4.21: Normalized power spectral density of test sequence (white Gaussian noise variance=25).

Three figures are shown next. They, in order, show the first, second and third order differentiation in time of the input sequence. As discussed in Chapter III, it is expected that the first order differentiation of the sequence should provide the best prewhitening. Figure 4.22 shows the averaged periodogram of the first order differentiation of the input sequence.

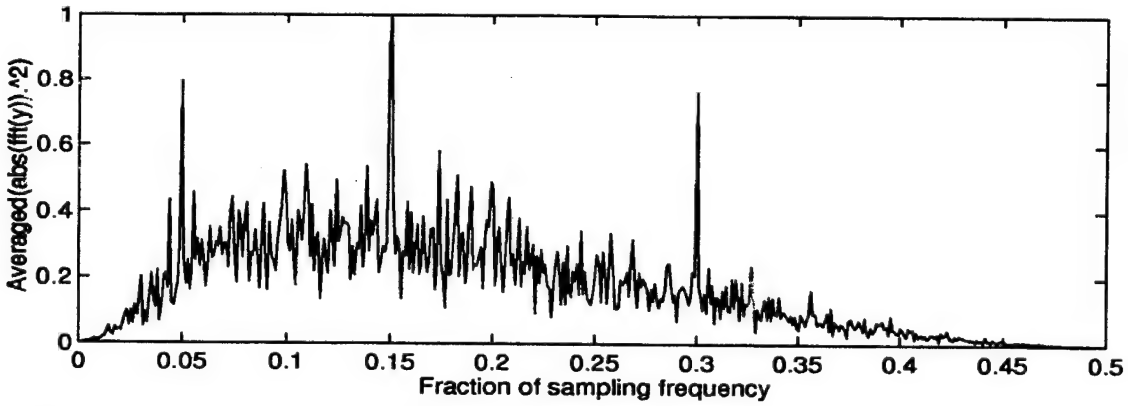

Figure 4.22: Normalized power spectral density of whitened sequence using first order differentiation.

As with some of the other whitening techniques, it can be readily observed that although the noise has been equalized somewhat, it still remains colored. Nevertheless, we can see that the three signal peaks are distinct and detectable. However, the form of the noise power spectral density would lead to better detection results in using a white noise approximation between the frequencies of 0.075 f_s - 0.2 f_s. In that area, the noise spectrum is relatively flat and signals such as Sinusoid #2 produce sharp easily detected peaks. Although Sinusoid #1 and Sinusoid #3 are not in areas where the noise power has leveled off, the local signal to noise ratio of all three sinusoids appears to be at least as high as the local signal to noise ratio shown on Figure 4.21. All three sinusoids of Figure 4.22 account for the global peaks whereas on Figure 4.21, Sinusoid #2 and Sinusoid #3 have less power than the noise in the low-pass region.

Because the noise is not completely equalized by using the first order differentiation, a second and third experiment are conducted to see what the effect of second order and third

order differentiation would have on the input sequence. Figure 4.23 shows the resulting averaged periodogram for the second order differentiation and Figure 4.24 contains the results of the third order differentiation.

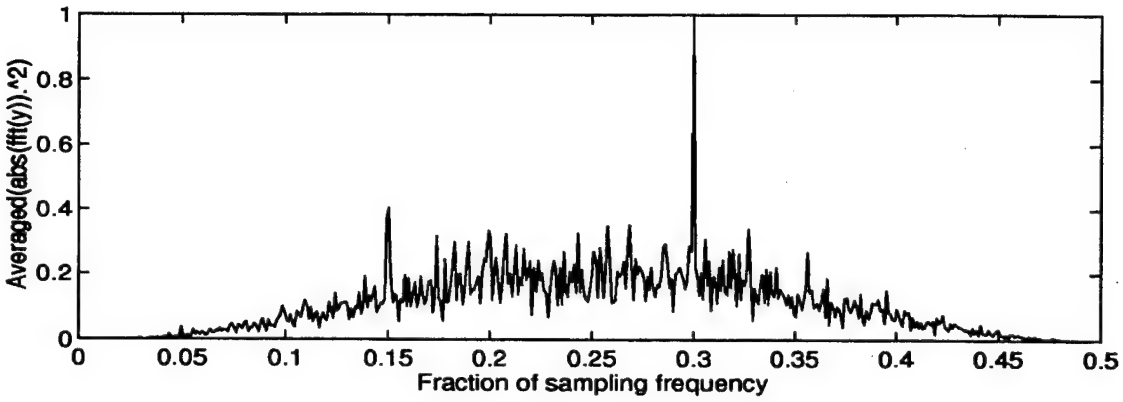

Figure 4.23: Normalized power spectral density of whitened sequence using second order differentiation.

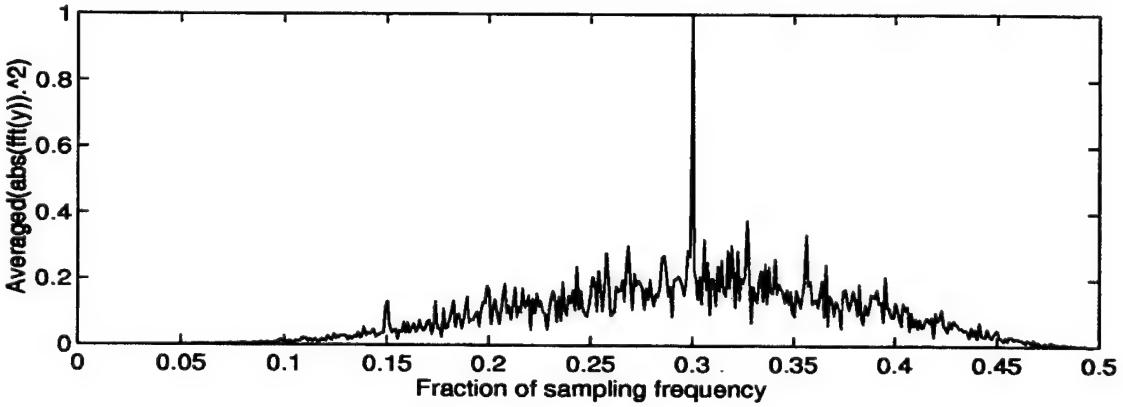

Figure 4.24: Normalized power spectral density of whitened signal using third order differentiation.

The effect of the additional orders of differentiation can be readily seen on the two figures. In all cases the noise power forms a characteristic hump which begins and ends with magnitudes close to or equal to zero. For the first order differentiation, the top of the hump is centered at a frequency of about 0.12 f_s. For the second order differentiation, the

hump moved to higher frequencies and its summit is around 0.26 f_s. In the case shown in Figure 4.24, the somewhat flat noise power area is in the frequency range of 0.2 f_s- 0.32 f_s and Sinusoid #3 at 0.3 f_s is clearly detectable. Sinusoid #1 at 0.05 f_s is far lower than the noise present at most frequencies and is not detectable. Sinusoid #2 is in the middle of the slope leading to the maximum noise power and has maintained its local signal to noise ratio but the noise at higher frequencies has higher power and would need to be filtered to allow for correct detection. For the third order differentiation, the hump moves to higher frequency with the top of the noise hump situated at a frequency around 0.32 f_s. Sinusoid #2 and Sinusoid #3 remain identifiable, but Sinusoid #1 has disappeared along with the noise in the frequencies around 0.05 f_s and lower. Because of its location in the flat portion of the noise spectrum Sinusoid #3 again is sharply defined and his easily detectable with a white noise assumption.

This method is a very easy and inexpensive way to minimize the effects of low frequency noise that has spectral roll-off as is found in the ocean environment.

2. Prediction-Error Filter (PEF)

This section demonstrates the whitening property of the prediction-error filter. As discussed in Chapter III, the success of the prediction error filter is predicated upon the correlation feature of the signal and of the noise. The signal used for this experiment is different than that used for the other experiments and is:

$$s[n] = 0.475\cos\left[2\pi n\frac{1}{8}\right] + 0.15\cos\left[2\pi n\frac{1}{4}\right] + 0.1\cos\left[2\pi n\frac{3}{8}\right] \qquad (4.5)$$

The signal is changed to better illustrate the performance of the filter and its dependence upon the correlation properties of the signal and colored noise. Figures 4.25 and 4.26 show the theoretical correlation functions of the colored noise and of the signal for the first 21 lags.

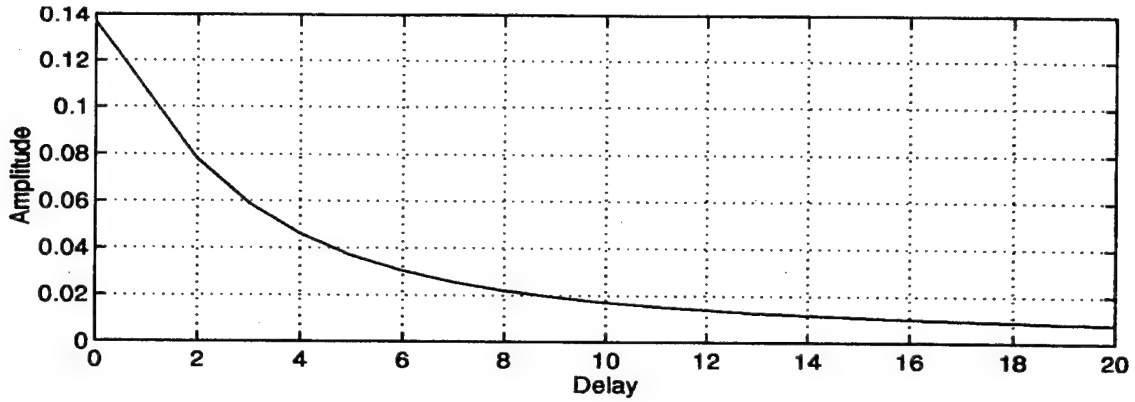

Figure 4.25: Autocorrelation function of the Butterworth low-pass noise: First 21 points.

Figure 4.25 shows that the magnitude of the autocorrelation function of the noise rapidly decreases with the number of delays. The autocorrelation function of underwater noise is very similar because it also has high noise power for low frequencies. This means that if a delay must be chosen to maximize the correlation of the noise, then this delay should be small. If the noise is white, the magnitude would be non-zero only at a delay of zero.

Figure 4.26 shows the autocorrelation function of the signal.

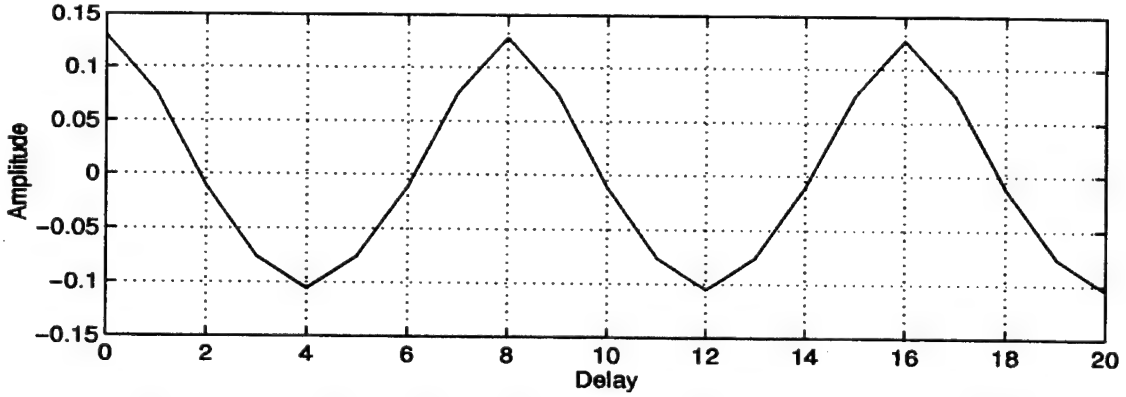

Figure 4.26: Autocorrelation function of the signal: s=.475*cos(2*pi*t/8) +0.15*cos(2*pi*t*2/8) + 0.1*cos(2*pi*t*3/8)

As expected, the autocorrelation function of the sinusoidal signal is also sinusoidal. We note that the signal is formed from three sinusoids each of which has its own correlation

function. The net results shown here presents the sum of all three autocorrelation functions. Therefore, if a delay is chosen where the autocorrelation function of the signal does not have a magnitude of zero as shown on Figure 4.26, each of the three sinusoids is affected differently based on the magnitude of their respective autocorrelation function at that delay.

In order to demonstrate the whitening properties of the prediction-error filter, an adaptive order 32 LMS filter was constructed. Data is input to the filter until the weights have converged and all results are computed only with data obtained after convergence. The whitened sequence is the error between the predicted value out of the predictor, and the actual value of the sequence. Therefore, if the signals are to be detected in the error, the delay d (see Figure 3.9) must be chosen such that the correlation of the noise is maximized while the correlation of the signal is minimized.

The next three figures present the results of the experiment. The Matlab code of Appendix Q was used to compute these results.

Figure 4.27 shows an averaged periodogram of the input sequence. The white Gaussian noise variance used to create the colored noise is 100. Experimentation shows that the correlation of the noise samples is most sensitive to delays. A delay of one achieved the best whitening. We expect that this is due to the fact that the colored noise has a correlation function which asymptotically decreases with delay. Therefore, as the delay is increased, the predictor quickly loses its ability to estimate the noise and therefore remove it from the error sequence. The signals can be seen 0.125 fs, 0.25 fs, and, 0.375 fs.

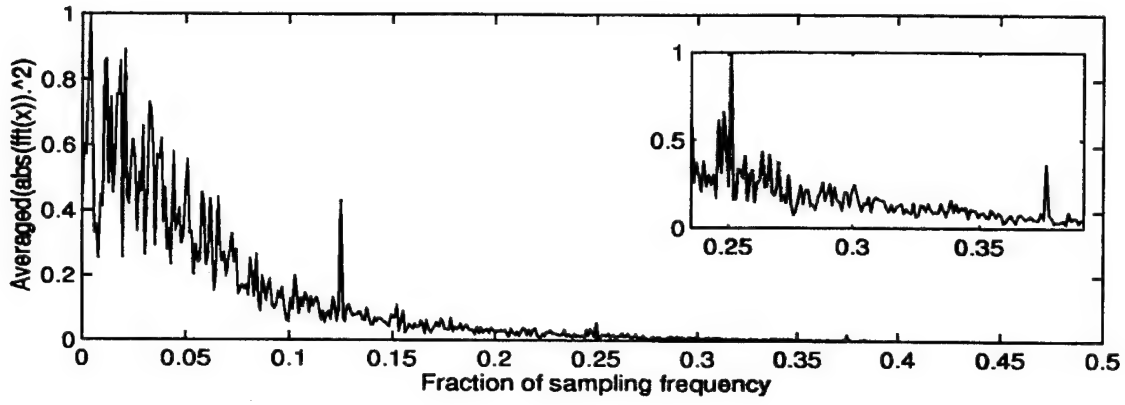

Figure 4.27: Normalized power spectral density of signal plus noise (white Gaussian noise variance=100).

Figure 4.28 shows the power spectral density of the output of the predictor.

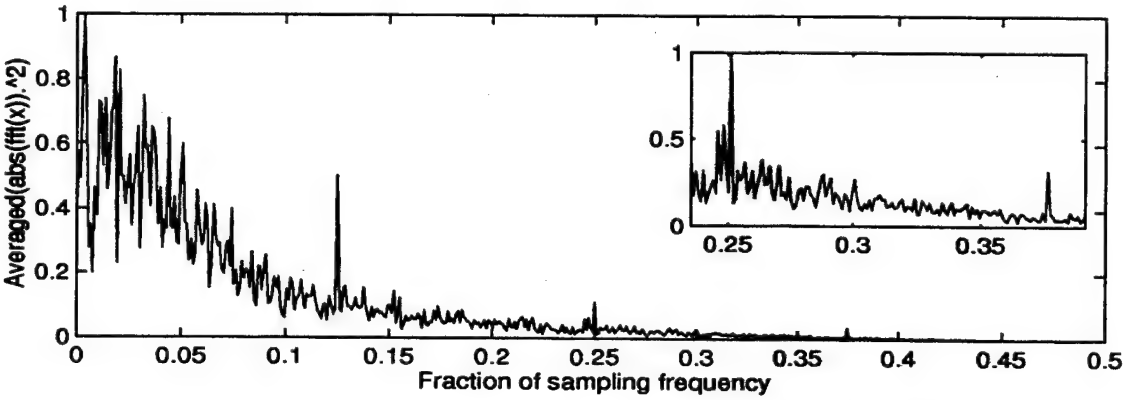

Figure 4.28: Normalized power spectral density of estimated signal.

For a delay of one, the autocorrelation of none of the three sinusoids is zero. Because of this, the predictor is also able to estimate their presence as well as that of the colored noise. Since the delay chosen is short, it can be expected that low frequency noise components and signals at low frequencies would be better predicted and therefore more easily removed from the whitened sequence or the error sequence. In the case of signals in low-pass noise, this is a good feature since the noise at low frequencies has far greater power than the signals which are at higher spectral locations. We note that the averaged

periodogram of the predicted signal looks very similar to the averaged periodogram of the input sequence.

Figure 4.29 shows the averaged periodogram of the error containing the whitened sequence.

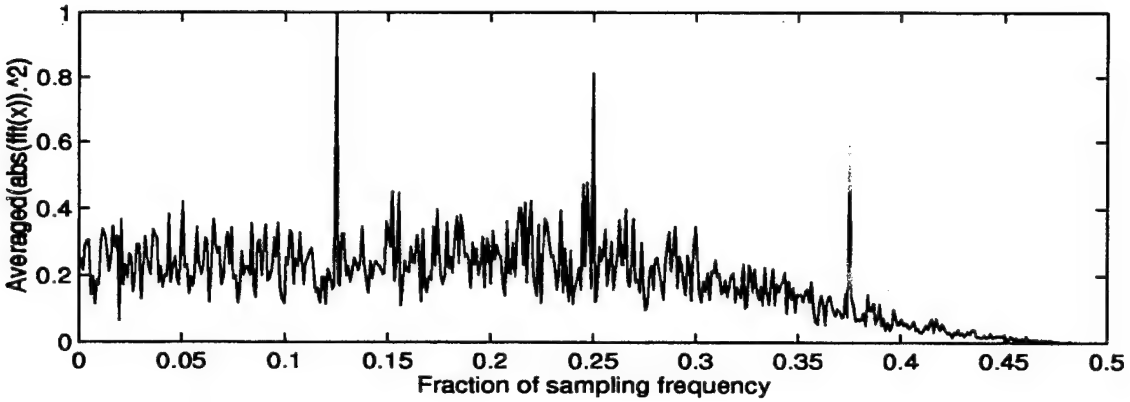

Figure 4.29: Normalized power spectral density of predicted error: whitened signal.

It can be seen that the noise has been considerably whitened by the PEF. At frequencies below $0.27 f_s$, the noise is effectively whitened. At higher frequencies, the small delay appears to have partly decorrelated some of the noise power, and some coloration remains. The three signals also appear clearly despite a delay that did not completely decorrelate them as indicated by their presence in the predicted sequence. Comparing Figures 4.27 and 4.29 shows that the ratios between the spectral heights of each sinusoid and the noise at their respective spectral location have been preserved. Different delays however, have a serious effect of the performance of this whitener. As the selected delay increases, the noise becomes more decorrelated, and the predictor is less able to produce correct estimates. This then results in more of the colored noise to be part of the error.

When this technique is used for detection in colored noise, its performance is dependent upon the correlation properties of the signal and noise. In the case of the noise, underwater ambient noise is always best predicted by choosing small delays. For the signal, which must be decorrelated as much as possible, a study of its correlation properties must

be undertaken to ensure that the delay chosen does not occur at a point where the signal autocorrelation function is at its maximum. Should this occur, the predictor is then completely able to predict the signal in its estimate and therefore none of the signal appears in the error sequence.

An advantage of the method presented herein, is that it is adaptive. As sea going vessels are underway, the noise characteristic of the underwater environment changes. It is reasonable to assume that this could affect our choice of delay. However, the variations in the ambient noise spectra usually only minimally affect this consideration, because the primary effect resulting in much higher noise levels at low frequency is the sound attenuation in sea water. Therefore using an adaptive filter in this manner allows these changes to be considered in the detection process without modifying the experimental parameters.

V. CONCLUSIONS

In this thesis, we have presented a number of techniques with which to perform discrete time detection of signals in colored noise. The matched filter for colored noise is derived and is shown to produce excellent results. When the correlation matrix of the noise can be estimated, the matched filter for colored noise provides the optimal detection solution. The derivation for the Maximum A Posteriori criterion for colored noise is also shown. The inverse filter is demonstrated to be a very efficient way to whiten the noise by effectively equalizing the colored noise by the use of its inverse power spectral density. This method is simple and effective with the only requirement being that the power spectral density of the noise be known. Several matrix decomposition techniques are addressed. These techniques are based on the use of a block transformation approach to whitening a sequence. They generally use a matrix of orthonormal vectors derived from the noise, to transform a data sequence in order to decorrelate the noise component of that sequence. If the colored noise can be decorrelated, then detection in white noise can be performed. The procedures to obtain a matrix of orthonormal vectors rely on a factorization of either the correlation, or covariance matrix of the colored noise process, or of a noise data matrix. The methods using the data matrix (i.e., signal free noise recording) are less susceptible to a poorly conditioned or singular correlation matrix which is necessarily estimated. They also result in better numerical accuracy which was found to be important for the whitening process. Some of the methods examined, for example, the LDU, UDL, Cholesky, and, QR factorizations, form a matrix which is triangular. Although some matrix decomposition methods require less processing resources than others, all of them remain very inefficient compared to other available techniques such as the matched filter and the inverse filter. Some of these techniques, although promising in theory, are sometimes unable to completely whiten the colored noise as predicted. For the low-pass Butterworth colored noise used, the eigenvector factorization method was unsuccessful in allowing for the detection of any of the three sinusoids. As a results of the poor conditioning of the noise

autocorrelation matrix, the transformation using the matrix of eigenvectors also whitens the signal along with the noise. Three methods work well in preserving the signal to noise ratio of the sinusoids while whitening the noise. The singular value decomposition computes a more accurate matrix of eigenvectors for the noise autocorrelation matrix and functions well in spite of poor conditioning of the autocorrelation matrix. It whitens the noise and generally maintains signal power. The QR technique also uses a noise data matrix to achieve a more precise estimate of the $\boldsymbol{L}$ matrix of the LDU factorization of the correlation matrix. As shown in Chapter 4, the QR method effectively decorrelates the colored noise and allows the use of an optimal white noise detector, to achieve optimal detection. The Cholesky factorization is able to correctly whiten the noise and maintain the appropriate spectral heights ratio but is not as capable as the QR factorization when the input noise power is greater. The differential operator can be a very efficient whitener when the colored noise is inversely proportional to a power of the frequency. The whitening property of the prediction-error filter is demonstrated and is shown to be effective. However, it is strongly dependent on the correlation properties of the signals to be detected. Some periodic signals may not be detected if the chosen delay occurs at a maximum of the signal correlation function. When the delay is at a point of high correlation for the noise, and at a point of low correlation for the signal, the prediction error filter can effectively whiten the noise and provide good output signal to noise ratios.

LIST OF REFERENCES

1. Kinsler, L.E., Frey, A.R., Coppens, A.B., Sanders, J.V., *Fundamentals of Acoustics, Third Edition*, John Wiley & Sons Inc., New York, NY, 1982.

2. Mellen, R.H., "The Thermal Noise Limit in the Detection of Underwater Acoustic Signals," *The Journal of the Acoustical Society of America*, 24:478, 1952.

3. Wenz, G.M., "Acoustic Ambient Noise in the Ocean," *The Journal of the Acoustical Society of America*, Vol. 34, No. 12, pp. 1936-1956, December 1962.

4. Urick, R.J., *Principles of Underwater Sound, 3rd Edition*, McGraw-Hill Book Company, New York, NY, 1983.

5. Efron, A.J., Jeen, H., *Detection in Impulsive Noise Based on Robust Whitening*, IEEE Transaction on Signal Processing, Vol 42, No 6, June 1994.

6. Rayleigh, Lord, *The Theory of Sound, Vol. II*, Dover publications Inc., New York, NY, 1945.

7. Leonard, R.W., Combs, P.C., Skidmore, L.R., "Attenuation of Sound in Synthetic Sea Water," *The Journal of the Acoustical Society of America*, 21:63, 1949.

8. Fisher, F.H., and Simmons, V.P., "Sound Absorption in sea water," *The Journal of the Acoustical Society of America*, Vol. 62, No. 3, pp. 558-564, September 1977.

9. Wilson, O.B, *Introduction to Theory and Design of Sonar Transducers*, Peninsula Publishing, Los Altos, CA, 1988.

10. Whalen, A.D., *Detection of Signals in Noise*, Academic Press Inc., San Diego, CA, 1971.

11. Van Tree, H.L., *Detection, Estimation, and Modulation Theory: Radar-Sonar Signal Processing and Gaussian Signals in Noise*, Original Edition, John Wiley & Sons, Inc., New York, NY, 1971, Reprint Edition, Krieger Publishing Company, Malabar, FL, 1992.

12. Hippenstiel, R. D., Introduction to Detection Theory, Teaching notes, Naval Postgraduate School.

13. Shanmugan, K.S., Breipohl, A.M., *Random Signals: Detection, Estimation and Data Analysis*, John Wiley & Sons, New York, NY, 1988.

14. Therrien, C.W., *Discrete Random Signals and Statistical Signal Processing*, Prentice Hall, Inc., Englewood Cliffs, NJ, 1992.

15. Proakis, J.G., Rader, C.M., Ling, F., Nikias, C.L., *Advanced Digital Signal Processing*, Macmillan Publishing Company, New York, NY, 1992.

16. Haykin, S., *Modern Filters*, Macmillan Publishing Company, New York, NY, 1989.

17. Golub, G.H., Van Loan, C.F., *Matrix Computations, Second Edition*, The John Hopkins University Press, Baltimore and London, 1989.

18. Haykin, S., *Adaptive Filter Theory, Second Edition*, Prentice Hall, Inc., Englewood Cliffs, NJ, 1991.

19. Proakis, J.G., Manolakis, D.G., *Digital Signal Processing: Principles, Algorithms, and Applications, Second Edition*, Macmillan Publishing Company, New York, NY, 1992.

APPENDIX A - MATLAB PROGRAM USED TO GENERATE FIGURE 2.2, "THERMAL AGITATION SOUND PRESSURE LEVEL (SPL)".

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plots thermal agitation noise levels
%
% Author:Capt M.A. Cloutier
% Date:   26 Oct 95
% Name:  thermalnoise.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clf

f=logspace(0,7,500);%frequency range

SPL=-101+20*log10(f);

semilogx(f,SPL)
axis([1 10^7 -100 40])
grid
xlabel('Frequency (Hz)')
ylabel('SPL (dB) re 2X10^3 uPa for a 1 Hz Bandwidth')

print thermalfig
!ps3epsi <thermalfig.ps >thermalfig.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX B - MATLAB PROGRAM USED TO GENERATE FIGURE 2.4, "COMPARISON OF ABSORPTION COEFFICIENTS IN SEA WATER, DUE TO SHEAR AND VOLUME VISCOSITY, AND SHEAR VISCOSITY ALONE".

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Thesis:  Plot of viscosity absorption coefficients iaw Fisher and Simmons
%salinity: 35 ppt
%PH=8.0
%ref page 104 Urick
%
% Author:Capt M.A. Cloutier
% Date:   26 Oct 95
% Name:  absorptionf21.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clf
f=logspace(0,7,500);%frequency range
T=5;        %temperature in Celcius
Po=1;       %pressure in atm
TK=T+273;%temp in Kelvin
f1=1.32*10^3*TK*exp(-1700/TK); %relaxation freq of boric acid
f2=1.55*10^7*TK*exp(-3052/TK);%relaxation freq of MgSO4

% for temps and pressures in ranges 0-30¡C and 1-400 atm
A=8.95*10^(-8)*(1+2.3*10^(-2)*T-5.1*10^(-4)*T^2);
B=4.88*10^(-7)*(1+1.3*10^(-2)*T)*(1-0.9*10^(-3)*Po);
C=4.76*10^(-13)*(1-4*10^(-2)*T+5.9*10^(-4)*T^2)*(1-3.8*10^(-4)*Po);

%attenuation coeff in dB/m
attboric=A*f1*f.^2./(f1^2+f.^2);
attMgSO4=B*f2*f.^2./(f2^2+f.^2);
attwater=C*f.^2;

att=attboric+attMgSO4+attwater;

mus=.01;%shear viscosity
muv=2.81*mus;%volume viscosity
density=1;%gm/cm3
c=1.5*10^5;%cm/s
```

```
%pure water
attshearvisc=16*pi^2*mus*f.^2./(3*density*c^3);
attvisc=16*pi^2*f.^2*(mus+0.75*muv)/(3*density*c^3);

loglog(f,att,'-',f,1000*attshearvisc,'--',f,attwater,'-.')
axis([1 10^7 10^(-10) 10^2])
ylabel('Absorption coefficient (dB/m)')
xlabel('Frequency (Hz)')

h=legend(' sea water',' shear and volume',' shear (theoretical)')
axes(h)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX C - MATLAB PROGRAM USED TO COMPUTE FIGURE 2.5, "SOUND ABSORPTION AT 5°C, 1 ATM, 35 PPT SALINITY, AND PH=8.0 (ACCORDING TO FISHER AND SIMMONS)".

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Thesis:  Plot of absorption coefficients according to Fisher and Simmons
%salinity: 35 ppt
%PH=8.0
%ref page 158 KFCS
% For second figure Chap 2
%
% Author:Capt M.A. Cloutier
% Date:   26 Oct 95
% Name:  absorptionf22.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clf
clear
f=logspace(0,7,500);%frequency range
T=5;%temperature in Celcius
Po=1;%pressure in atm

TK=T+273;%temp in Kelvin
f1=1.32*10^3*TK*exp(-1700/TK); %relaxation freq of boric acid
f2=1.55*10^7*TK*exp(-3052/TK);%relaxation freq of MgSO4

% for temps and pressures in ranges 0-30¡C and 1-400 atm

A=8.95*10^(-8)*(1+2.3*10^(-2)*T-5.1*10^(-4)*T^2);
B=4.88*10^(-7)*(1+1.3*10^(-2)*T)*(1-0.9*10^(-3)*Po);
C=4.76*10^(-13)*(1-4*10^(-2)*T+5.9*10^(-4)*T^2)*(1-3.8*10^(-4)*Po);

%attenuation coeff in dB/m

attboric=A*f1*f.^2./(f1^2+f.^2);
attMgSO4=B*f2*f.^2./(f2^2+f.^2);
attwater=C*f.^2;

att=attboric+attMgSO4+attwater;
```

```
loglog(f,att,'-',f,attwater,'--',f,attboric,':',f,attMgSO4,'-.')
axis([1 10^7 10^(-10) 10^2])
ylabel('Absorption coefficient (dB/m)')
xlabel('Frequency (Hz)')
h=legend(' sea water',' fresh water',' boric acid',' magnesium sulfate')
axes(h)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX D - MATLAB PROGRAM USED TO GENERATE FIGURE 2.7, "NORMALIZED AVERAGE SOUND PRESSURE LEVEL OF AMBIENT NOISE AT POINT SUR SOSUS ARRAY (SAMPLING FREQUENCY 8 KHZ)".

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PSD of ocean noise from surnoise file taken at 8 kHz on a log scale
% From surnoise.au
%
% Author:Capt M.A. Cloutier
% Date:   26 Oct 95
% Name: noisesurPSDlogpad.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
clf
orient tall
acnoise=auread('surnoise.au');
%nacnoise=acnoise-mean(acnoise);%remove DC
index=1;
for i=1:2048:40960
   PSD(:,index)=20*log10(abs(fft(acnoise(i:i+2047),8192)));
   index=index+1;
end
PSDave=sum(PSD')./(index-1);
%plot in semilog space
clf
subplot(211)
point=logspace(0,log10(4096),500);
semilogx(point*8000./8192,PSDave(point)-max(PSDave(point)))
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')

axis([10^1 10^4 min(PSDave(point)-max(PSDave(point))) 0])

print fsurnoisePSD
!ps3epsi <fsurnoisePSD.ps >fsurnoise.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX E - MATLAB PROGRAM USED TO GENERATE FIGURE 3.4, "DETECTION REGIONS: GAUSSIAN PROBABILITY DENSITY FUNCTIONS".

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program that plots detection regions
%
% Author:Capt M.A. Cloutier
% Date:   26 Oct 95
% Name:  gauss.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
orient tall
i=-15:.1:25;
m=10;
var=4;
p1=1/(sqrt(2*pi*var^2))*exp(-(i-m).^2./(2*var^2));
p0=1/(sqrt(2*pi*var^2))*exp(-(i).^2./(2*var^2));

T=linspace(0,0.75/(sqrt(2*pi*var^2)),20);
j=5*ones(size(T));

subplot(311)

plot(i,p1,i,p0,j,T)

hold on

fill([i(202) i(202:401)],[0 p1(202:401)],'m');
fill([i(202) i(202:401)],[0 p0(202:401)],'g');

fill([i(1:200) i(200)],[p0(1:200) 0],'c');
fill([i(1:200) i(200)],[p1(1:200) 0],'y');

axis('off')
print gaussfig

!ps3epsi <gaussfig.ps >gaussfig.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX F - MATLAB PROGRAM USED TO GENERATE FIGURE 3.10, "CORRELATION FUNCTION OF FIRST ORDER LOW-PASS BUTTERWORTH COLORED NOISE".

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program to compute the theoretical autocorr of noise from the PSD
%
% Author:Capt M.A. Cloutier
% Date:   2 Dec 95
% Name:  wienerk.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clf
orient tall

l=1024;
t=1:l;
s= 1*sin(2*pi*t*1/4);%f is at 0.25 fs

%varia=input('What is the variance of the white noise? ');
varia=1;
w=sqrt(varia)*randn(1,length(s));%white noise

[B,A]=butter(1,0.1);%color the white noise
n=filter(B,A,w);

x=s+n;

PSD=abs(fft(n,2048)).^2./2048;%PSD of the noise

[H,P]=freqz(B,A,2047);%H contains the transfer function

Rnt=real(ifft(abs([H.' fliplr(H.')]).^2));%with Wiener-Khinchine theorem find correlation matrix

Rn=xcorr(n); %has length 2*l-1
Rn1=Rn.';%form row vector instead of column vector
%form new Rn with peak at zero instead of centered
Rn2=[Rn1(1024:2047) Rn1(1:1023)];
impulse=dimpulse(B,A,1024);
Rnti=xcorr(impulse);
```

```
Rnmat=Rnti(1024:2047);
subplot(411)
plot(0:1023,Rn2(1:1024)/1024,'-', 0:1023,Rnt(1:1024),'.')
title('Plots of Rn with XCORR vs Rn with Wiener-Khinchine')

subplot(412)
plot(0:19,Rn2(1:20)/1024, 0:19, Rnt(1:20),0:19,Rnti(1024:1043))
title('Plots of first 20 points of correlations')

subplot(413)
plot(abs(fft(Rn2/2047)))
title('PSD with Rn2')

subplot(414)
plot(abs(fft(Rnt)))
title('PSD with Rnt')
print figwk
pause
clf

%for thesis chap4
subplot(411)
plot(0:25,Rn2(1:26)/1024,'-',0:25,zeros(1,26),'.')
axis([0 25 -.05 .15])
xlabel('Delay')
ylabel('Amplitude')
print figcorr
!ps3epsi <figcorr.ps >figcorr.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX G - MATLAB PROGRAM USED TO GENERATE FIGURE 4.1, "NORMALIZED AVERAGE POWER SPECTRAL DENSITY OF SIGNAL PLUS NOISE".

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This will create the signal plus noise signal
% and will produce the thesis figure in Chap 4
%
% Author:Capt M.A. Cloutier
% Date:   26 Oct 95
% Name: signalf.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clear
clf
orient landscape

% generate data
numb=100;%number of realizations for the averaging
l=numb*1024;
t=1:l;
s= .475*cos(2*pi*t/20) + 0.2*cos(2*pi*t*3/20) + 0.07*cos(2*pi*t*3/10 + pi/3);
%varia=input('What is the variance of the white noise? ');
varia=100;
w=sqrt(varia)*randn(1,length(s));%white noise

[B,A]=butter(1,0.1);%color the white noise
colored=filter(B,A,w);

x=s+colored;

PSD=zeros(numb,2048);
index=1;
for i=1:numb
   PSD(i,:)=abs(fft(x(index:(index+1023)),2048)).^2./2048;
   index=index+1024;
end

PSDxave=sum(PSD)./numb;
```

```
PSDxave=PSDxave./max(PSDxave);%normalize

%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output results

plot((1:1024)/1024/2,PSDxave(1:1024));
xlabel('Fraction of sampling frequency')
ylabel('Average(abs(fft(x,2048)).^2)')

text(.08,.95*max(PSDxave),'Sin #1')

axes('position',[.4 .35 .15 .5])
plot((261:360)/1024/2,PSDxave(261:360))
title('Sin #2')
axis([261/2/1024 360/2/1024 0 .2])
grid

axes('position',[.7 .35 .15 .5])
plot((571:670)/1024/2,PSDxave(571:670))
title('Sin #3')
axis([571/2/1024 670/2/1024 0 .025])
grid

print sigffig1
!ps3epsi <sigffig1.ps >sigffig.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX H - MATLAB PROGRAM IMPLEMENTING THE DISCRETE TIME MATCHED FILTER FOR COLORED NOISE

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Discrete Matched Filter for colored noise
%
% Author:Capt M.A. Cloutier
% Date:   5 Dec 1995
% Name: mffin.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
orient tall

% generate data
numb=10;%number of realizations for the averaging
l=numb*1024;
t=1:l;
s= .475*cos(2*pi*t/20) + 0.2*cos(2*pi*t*3/20) + 0.07*cos(2*pi*t*3/10 + pi/3);
%varia=input('What is the variance of the white noise? ');
varia=100;
w=sqrt(varia)*randn(1,length(s));%white noise

[B,A]=butter(1,0.1);%color the white noise
colored=filter(B,A,w);

x=s+colored;

x=reshape(x,1024,numb)';
colored=reshape(colored,1024,numb)';

% Find correlation matrix of the noise from PSD

[H,P]=freqz(B,A,2047);%H contains the transfer function

Rn=real(ifft(abs([H.' fliplr(H.')]).^2));%with Wiener-Khinchine theorem find
correlation matrix

Rntoep=toeplitz(Rn(1:128));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculate h
```

```
h=inv(Rntoep)*fliplr(s(1:128)')./sqrt(s(1:128)*inv(Rntoep)*s(1:128)');
y=zeros(numb,1024);
PSDy=zeros(numb,1024);
PSDx=zeros(numb,1024);
for i=1:numb
   for j=1:(1024-127)
         y(i,j)=h.'*fliplr(x(i,j:(j+127)))';
   end
   PSDy(i,:)=(abs(fft(y(i,:),1024)).^2);
   PSDx(i,:)=(abs(fft(x(i,:),1024)).^2);
end

PSDyave=sum(PSDy)./numb;
PSDyave=PSDyave/max(PSDyave);

PSDxave=sum(PSDx)./numb;
PSDxave=PSDxave/max(PSDxave);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%output the estimates
clf
subplot(311)

plot((0:511)./1024, PSDxave(1:512))
ylabel('Averaged(abs(fft(x)).^2)')
xlabel('Fraction of sampling frequency')

axes('position',[.45 .78 .15 .12])
plot((130:180)/1024,PSDxave(130:180)./max(PSDxave(130:180)))
axis([130/1024 180/1024 0 1])

axes('position',[.7 .78 .15 .12])
plot((285:335)/1024,PSDxave(285:335)/max(PSDxave(285:335)))
axis([285/1024 335/1024 0 1])

print mffigx

!ps3epsi <mffigx.ps >mffigx.epsi

clf
subplot(311)
```

```
plot((0:511)./1024, PSDyave(1:512))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

print mffigy
!ps3epsi <mffigy.ps >mffigy.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX I - MATLAB PROGRAM IMPLEMENTING THE INVERSE FILTER

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Program to compute coloring and whitening of a sequence with white noise
% using an inverse filter
%
% Author:Capt M.A. Cloutier
% Date:   26 Oct 95
% Name:  invfin.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
orient tall

numb=10;
P=zeros(numb,512);
Pxa=zeros(numb,512);

for ind=1:numb

   l=1024;
   t=1:l;
   s=.475*cos(2*pi*t/20) + 0.2*cos(2*pi*t*3/20) + 0.07*cos(2*pi*t*3/10);
   varia=25;
   w=sqrt(varia)*randn(1,length(s));%white noise

   [B,A]=butter(1,0.1);   %color the white noise
   n=filter(B,A,w);

   x=s+n;

   white=filter(A,B,x);

   PSDn=abs(fft(x)).^2;

   PSDw=abs(fft(white)).^2;

   P(ind,:)=PSDw(1:512);
   Pxa(ind,:)=PSDn(1:512);

end
```

```
Pyave=sum(P)./numb;
Pxave=sum(Pxa)./numb;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output results

clf
subplot(311)
plot((0:511)./1024,Pxave./max(Pxave))
ylabel('Averaged(abs(fft(x)).^2)')
xlabel('Fraction of sampling frequency')

axes('position',[.45 .78 .15 .12])
plot((130:180)/1024,Pxave(130:180)./max(Pxave(130:180)))
axis([130/1024 180/1024 0 1])

axes('position',[.7 .78 .15 .12])
plot((285:335)/1024,Pxave(285:335)/max(Pxave(285:335)))
axis([285/1024 335/1024 0 1])

print invfigx

!ps3epsi <invfigx.ps >invfigx.epsi

clf
subplot(311)
plot((0:511)./l,Pyave./max(Pyave))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

print invfigy
!ps3epsi <invfigy.ps >invfigy.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX J - MATLAB PROGRAM IMPLEMENTING THE EIGENVECTOR FACTORIZATION

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Diagonalization of the Correlation matrix of the noise
% ref p50, 2.6.1
% do transformation y=E*T*x with E*T of the noise
% Use theoretical Rn estimated from PSD
%
% Author:Capt M.A. Cloutier
% Date:   2 Dec 95
% Name: eigfactfin.m
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clear
clf
orient tall
%
% generate data
numb=10;
PSDxa=zeros(numb,512);
PSDya=zeros(numb,512);

for ind2=1:numb
  l=1024;
  t=1:l;
  s= .475*cos(2*pi*t/20) + 0.2*cos(2*pi*t*3/20) + 0.07*cos(2*pi*t*3/10);

  %varia=input('What is the variance of the white noise? ');
  varia=10;
  w=sqrt(varia)*randn(1,length(s));%white noise

  [B,A]=butter(1,0.1);%color the white noise
  n=filter(B,A,w);

  x=s+n;

  %%%%%%%%%%%%%%%%%%%%%%%%%%%
  %calculate autocorrelation sequence of the noise
```

```
[H,P]=freqz(B,A,2048);%H contains the transfer function

Rnt=real(ifft(abs([H.' fliplr(H.')]).^2));%with Wiener-Khinchine theorem find correlation matrix
Rnmat=toeplitz(Rnt(1:256));

[V,D]=eig(Rnmat);% D:diagonal matrix of eigenvalue,V eigenvectors x*V=V*D

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
index=1;

%do transformation of x in blocks of 256 points

for i=1:4
        y(index:index+255)=V'*x(index:index+255).';
        index=index+256;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Calculate average PSD for x and y

PSDx=abs(fft(x,1024).^2);
PSDxa(ind2,:)=PSDx(1:512);

PSDy=abs(fft(y,1024)).^2;
PSDya(ind2,:)=PSDy(1:512);

end

PSDyave=sum(PSDya)./numb;
PSDxave=sum(PSDxa)./numb;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Output results

clf
subplot(311)
plot((0:511)/1024,PSDxave(1:512)./max(PSDxave(1:512)));
ylabel('Averaged(abs(fft(x)).^2)')
xlabel('Fraction of sampling frequency')

axes('position',[.45 .78 .15 .12])
plot((130:180)/1024,PSDxave(130:180)./max(PSDxave(130:180)))
```

```
axis([130/1024 180/1024 0 1])

axes('position',[.7 .78 .15 .12])
plot((285:335)/1024,PSDxave(285:335)/max(PSDxave(285:335)))
axis([285/1024 335/1024 0 1])

print eigfactfigx

!ps3epsi <eigfactfigx.ps >eigfactfigx.epsi

clf
subplot(311)
plot((0:511)/1024,PSDyave(1:512)./max(PSDyave(1:512)));
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

print eigfactfigy
!ps3epsi <eigfactfigy.ps >eigfactfigy.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX K - MATLAB PROGRAM IMPLEMENTING THE SINGULAR VALUE DECOMPOSITION

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Single Value Decomposition:  From singvdbigN.m
%
% Author:Capt M.A. Cloutier
% Date:   15 Nov 95
% Name: svdfin.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
clf
% generate data
l=1024;
t=1:l;
s=.475*cos(2*pi*t/20) + 0.2*cos(2*pi*t*3/20) + 0.07*cos(2*pi*t*3/10 + pi/3);
varia=50;
PSDy=zeros(10,1024);
PSDx=zeros(10,1024);
average=20;

for j=1:average
   w=sqrt(varia)*randn(1,l);%white noise
   [B,A]=butter(1,0.1);   %color the white noise
   n=filter(B,A,w);

   x=s+n;

   %Define matrix of sample vectors of the noise:  Use 48 vectors 1024 pts long

   for i=1:48
         N(i,:)=n(1:1024);
         w=sqrt(varia)*randn(1,1024);%white noise
         n=filter(B,A,w);
   end
   %Perform the SVD on the noise data matrix N
   % N=UEVh: where V is the matrix of eigenvectors

   [U,S,V] = svd(N);

   %do transformation in one block of 1024 points
```

```
   index=1;
   y(index:index+1023)=V'*x(index:index+1023).';

   PSDy(j,:)=abs(fft(y,1024)).^2;
   PSDx(j,:)=abs(fft(x,1024)).^2;
end

PSDyave=sum(PSDy(:,1:512))./average;
PSDxave=sum(PSDx(:,1:512))./average;

clf
subplot(311)
plot((0:511)./1024,PSDxave./max(PSDxave))
ylabel('Averaged(abs(fft(x)).^2)')
xlabel('Fraction of sampling frequency')

axes('position',[.45 .78 .15 .12])
plot((130:180)/1024,PSDxave(130:180)./max(PSDxave(130:180)))
axis([130/1024 180/1024 0 1])

axes('position',[.7 .78 .15 .12])
plot((285:335)/1024,PSDxave(285:335)/max(PSDxave(285:335)))
axis([285/1024 335/1024 0 1])

print svdfigx

!ps3epsi <svdfigx.ps >svdfigx.epsi

clf
subplot(311)
plot((0:511)./l,PSDyave./max(PSDyave))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

print svdfigy
!ps3epsi <svdfigy.ps >svdfigy.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX L - MATLAB PROGRAM IMPLEMENTING THE MAHALANOBIS TRANSFORMATION

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mahalanobis Transformation using the Correlation matrix of the noise
% ref p247, eq 5.63
% do transformation y=Rn^-.5*x with Rn of the noise
%
% Author:Capt M.A. Cloutier
% Date:   5 Dec 95
% Name:  mahafin.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
clf
orient tall
numb=10;
PSDx=zeros(numb,512);
PSDy=zeros(numb,512);

for ind=1:numb
   % generate data
   l=1024;
   t=1:l;
   s= .475*cos(2*pi*t/20) + 0.2*cos(2*pi*t*3/20) + 0.07*cos(2*pi*t*3/10);
   varia=25;
   w=sqrt(varia)*randn(1,length(s));%white noise
   [B,A]=butter(1,0.1);   %color the white noise
   n=filter(B,A,w);

   x=s+n;

   [H,P]=freqz(B,A,256);%H contains the transfer function
   Rnt=real(ifft(abs([H.' fliplr(H.')]).^2));%with  Wiener-Khinchine  theorem  find
correlation matrix
   Rnmat=toeplitz(Rnt(1:256));

   index=1;
   %do transformation in blocks of 256 points
   for i=1:4
         y(index:index+255)=inv(sqrt(Rnmat))*x(index:index+255).';
         index=index+256;
```

```
    end

    Py=abs(fft(y,1024)).^2;
    Px=abs(fft(x,1024)).^2;
    PSDy(ind,:)=Py(1:512);
    PSDx(ind,:)=Px(1:512);
end
Pyave=sum(PSDy)./numb;
Pxave=sum(PSDx)./numb;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output results
clf
subplot(311)
plot((0:511)./1024,Pxave./max(Pxave))
ylabel('Averaged(abs(fft(x)).^2)')
xlabel('Fraction of sampling frequency')
axes('position',[.45 .78 .15 .12])
plot((130:180)/1024,Pxave(130:180)./max(Pxave(130:180)))
axis([130/1024 180/1024 0 1])
axes('position',[.7 .78 .15 .12])
plot((285:335)/1024,Pxave(285:335)/max(Pxave(285:335)))
axis([285/1024 335/1024 0 1])
print mahafigx
!ps3epsi <mahafigx.ps >mahafigx.epsi

clf
subplot(311)
plot((0:511)./1024,Pyave./max(Pyave))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')
axes('position',[.17 .78 .12 .12])
plot((25:75)/1024,Pyave(25:75)/max(Pyave(25:75)))
axis([25/1024 75/1024 0 1])

print mahafigy
!ps3epsi <mahafigy.ps >mahafigy.epsi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX M - MATLAB PROGRAM IMPLEMENTING THE LDU AND UDL TRIANGULAR FACTORIZATIONS

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Diagonalization by triangular decomposition using LDU and UDL decomp
% LDL: y=inv(L)*x ; UDL: y=inv(Lu)*x
%
% Author:Capt M.A. Cloutier
% Date:   5 Dec 95
% Name: LDLfin.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ref p64, 2.7.1
%
clear
clf
orient tall

numb=10;
PSDx=zeros(numb,512);
PSDy=zeros(numb,512);

for ind=1:numb

   % generate data
   l=1024;
   t=1:l;
   s= .475*cos(2*pi*t/20) + 0.2*cos(2*pi*t*3/20) + 0.07*cos(2*pi*t*3/10);
   varia=25;
   w=sqrt(varia)*randn(1,length(s));%white noise

   [B,A]=butter(1,0.1);   %color the white noise
   n=filter(B,A,w);

   x=s+n;

   [H,P]=freqz(B,A,2047);%H contains the transfer function

   Rnt=real(ifft(abs([H.' fliplr(H.')]).^2));%with   Wiener-Khinchine   theorem   find
correlation matrix
   Rnmat=toeplitz(Rnt(1:256));
```

```
    [L,U,P]=lu(Rnmat);%for LDU

    [Lu,Uu,Pu]=lu(fliplr(flipud(Rnmat)));%UDL:LDU or reversed Rn

    %do transformation y=inv(L)x

    index=1;

    for i=1:4
            y(index:(index+255))=inv(L)*x(index:(index+255)).';
            yh(index:(index+255))=inv(fliplr(flipud(Lu)))*x(index:(index+255)).';

            index=index+256;
    end

    Py=abs(fft(y,1024)).^2;
    Pyh=abs(fft(yh,1024)).^2;

    Px=abs(fft(x,1024)).^2;

    PSDy(ind,:)=Py(1:512);
    PSDyh(ind,:)=Pyh(1:512);
    PSDx(ind,:)=Px(1:512);

end

Pyave=sum(PSDy)./numb;
Pyaveh=sum(PSDyh)./numb;

Pxave=sum(PSDx)./numb;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output results

clf
subplot(311)
plot((0:511)./1024,Pxave./max(Pxave))
ylabel('Averaged(abs(fft(x)).^2)')
xlabel('Fraction of sampling frequency')

axes('position',[.45 .78 .15 .12])
plot((130:180)/1024,Pxave(130:180)./max(Pxave(130:180)))
axis([130/1024 180/1024 0 1])
```

```
axes('position',[.7 .78 .15 .12])
plot((285:335)/1024,Pxave(285:335)/max(Pxave(285:335)))
axis([285/1024 335/1024 0 1])

print ldlfigx
!ps3epsi <ldlfigx.ps >ldlfigx.epsi

clf
subplot(311)
plot((0:511)./1024,Pyave./max(Pyave))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

print ldlfigy
!ps3epsi <ldlfigy.ps >ldlfigy.epsi

clf
subplot(311)
plot((0:511)./1024,Pyaveh./max(Pyaveh))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

print ldlhfigy
!ps3epsi <ldlhfigy.ps >ldlhfigy.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX N - MATLAB PROGRAM IMPLEMENTING THE CHOLESKY FACTORIZATION

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Diagonalization by triangular decomposition: Cholesky
% y=inv(L)*x (upper) and y=inv(fliplr(flipud(Ru')))*x (lower)
%
% Author:Capt M.A. Cloutier
% Date:   5 Dec 95
% Name:  cholfin.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clear
clf
orient tall

numb=10;
PSDx=zeros(numb,512);
PSDy=zeros(numb,512);

for ind=1:numb
  % generate data
  l=1024;
  t=1:l;
  s= .475*cos(2*pi*t/20) + 0.2*cos(2*pi*t*3/20) + 0.07*cos(2*pi*t*3/10 + pi/3);
  varia=25;
  w=sqrt(varia)*randn(1,length(s));%white noise

  [B,A]=butter(1,0.1);   %color the white noise
  n=filter(B,A,w);

  x=s+n;

  [H,P]=freqz(B,A,2047);%H contains the transfer function

  Rnt=real(ifft(abs([H.' fliplr(H.')]).^2));%with  Wiener-Khinchine  theorem  find
correlation matrix

  Rnmat=toeplitz(Rnt(1:256));

  R=chol(Rnmat);        % for lower-upper
```

```
    Ru=chol(fliplr(flipud(Rnmat)));% for upper lower

    %do transformation y=inv(R')x

    index=1;
                                %The matlab chol produces an upper triangular
    for i=1:4
        y(index:(index+255))=inv(R')*x(index:(index+255)).';
        yh(index:(index+255))=inv(fliplr(flipud(Ru')))*x(index:(index+255)).';
        index=index+256;
    end

    Py=abs(fft(y,1024)).^2;
    Pyh=abs(fft(yh,1024)).^2;
    Px=abs(fft(x,1024)).^2;

    PSDy(ind,:)=Py(1:512);
    PSDyh(ind,:)=Pyh(1:512);
    PSDx(ind,:)=Px(1:512);

end

Pyave=sum(PSDy)./numb;
Pyaveh=sum(PSDyh)./numb;

Pxave=sum(PSDx)./numb;

%%%%%%%%%%%%%%%%%%%%%%%%
% Output results
clf
subplot(311)
plot((0:511)./1024,Pxave./max(Pxave))
ylabel('Averaged(abs(fft(x)).^2)')
xlabel('Fraction of sampling frequency')

axes('position',[.45 .78 .15 .12])
plot((130:180)/1024,Pxave(130:180)./max(Pxave(130:180)))
axis([130/1024 180/1024 0 1])

axes('position',[.7 .78 .15 .12])
plot((285:335)/1024,Pxave(285:335)/max(Pxave(285:335)))
axis([285/1024 335/1024 0 1])
```

```
print cholfigx
!ps3epsi <cholfigx.ps >cholfigx.epsi

clf
subplot(311)
plot((0:511)./1024,Pyave./max(Pyave))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

print cholfigy
!ps3epsi <cholfigy.ps >cholfigy.epsi

clf
subplot(311)
plot((0:511)./1024,Pyaveh./max(Pyaveh))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

print cholhfigy
!ps3epsi <cholhfigy.ps >cholhfigy.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX O - MATLAB PROGRAM IMPLEMENTING THE QR FACTORIZATION

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Diagonalization by triangular decomposition
%
% QR: y=inv(L)*x
% QR uses a data matrix:The QR factorization provides the factors in the
% triangular decomposition of the correlation matrix.
% X=QR: data matrix X is expressed as the product of a rectangular matrix
% whose columns are orthonormal and a square upper triangular matrix.
%
% This version uses a 256X256 noise data matrix
%
% Author:Capt M.A. Cloutier
% Date:   26 Oct 95
% Name:  QRbigfin.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clear
clf
orient tall

numb=10;
PSDx=zeros(numb,512);
PSDy=zeros(numb,512);

for ind=1:numb

  % generate data
  l=1024;
  t=1:l;
  s= .475*cos(2*pi*t/20) + 0.2*cos(2*pi*t*3/20) + 0.07*cos(2*pi*t*3/10 + pi/3);
  varia=50;

  w=sqrt(varia)*randn(1,256*256);%white noise

  [B,A]=butter(1,0.1);   %color the white noise
  n=filter(B,A,w);

  x=s+n(1:1024);
```

```
%Define matrix of sample vectors of the noise:  Use 256 vectors 256 pts long

N=zeros(256,256);

index=1;
for i=1:256
        N(i,:)=n(index:(index+255));
        index=index+256;
end

[Q,R]=qr(N);

Dhalf=(1/sqrt(256))*diag(diag(R));

L=(1/sqrt(256))*R'*inv(Dhalf);

%do transformation y=inv(L)x
%do transformation in blocks of 256 points
index =1;
for i=1:4
        y(index:index+255)=inv(L)*x(index:index+255).';
        index=index+256;
end

Py=abs(fft(y,1024)).^2;
Px=abs(fft(x,1024)).^2;

PSDy(ind,:)=Py(1:512);
PSDx(ind,:)=Px(1:512);

end

Pyave=sum(PSDy)./numb;
Pxave=sum(PSDx)./numb;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output results

clf
subplot(311)
plot((0:511)./1024,Pxave./max(Pxave))
ylabel('Averaged(abs(fft(x)).^2)')
```

```
xlabel('Fraction of sampling frequency')

axes('position',[.45 .78 .15 .12])
plot((130:180)/1024,Pxave(130:180)./max(Pxave(130:180)))
axis([130/1024 180/1024 0 1])

axes('position',[.7 .78 .15 .12])
plot((285:335)/1024,Pxave(285:335)/max(Pxave(285:335)))
axis([285/1024 335/1024 0 1])

print QR2figx
!ps3epsi <QR2figx.ps >QR2figx.epsi

clf
subplot(311)
plot((0:511)./1024,Pyave./max(Pyave))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

print QR2figy
!ps3epsi <QR2figy.ps >QR2figy.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX P - MATLAB PROGRAM IMPLEMENTING THE DIFFERENTIAL WHITENING FILTER

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Noise Whitener based on differentiation
%
% Author:Capt M.A. Cloutier
% Date:   29 Nov 95
% Name:  differefin.m
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
clf
orient tall

numb=10;
PSDx=zeros(numb,512);
PSDy=zeros(numb,512);
PSDy2=zeros(numb,512);
PSDy3=zeros(numb,512);

%%%%%%%%%%%%%%%%%%%%%%%%%
for ind=1:numb

  l=1024;
  t=1:l;

  s= .475*cos(2*pi*t/20) + 0.2*cos(2*pi*t*3/20) + 0.07*cos(2*pi*t*3/10 + pi/3);
  varia=25;

  w=sqrt(varia)*randn(1,length(s));%white noise

  [B,A]=butter(1,0.1);   %color the white noise
  n=filter(B,A,w);

  x=s+n;

  %%%%%%%%%%%%%%%%%%%%%%%%%%%
  % First order differentiation

  y=diff(x);
```

```
    Py=abs(fft(y,1024)).^2;
    Px=abs(fft(x,1024)).^2;

    PSDy(ind,:)=Py(1:512);
    PSDx(ind,:)=Px(1:512);

    %%%%%%%%%%%%%%%%%%%%%%%%%%
    % Second order differentiation

    y2=diff(x,2);

    Py2=abs(fft(y2,1024)).^2;

    PSDy2(ind,:)=Py2(1:512);

    %%%%%%%%%%%%%%%%%%%%%%%%%%
    % Third order differentiation

    y3=diff(x,3);

    Py3=abs(fft(y3,1024)).^2;

    PSDy3(ind,:)=Py3(1:512);

end

Pyave=sum(PSDy)./numb;
Pxave=sum(PSDx)./numb;

%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output results
%%%%%%%%%%%%%%%%%%%%%%%%%%
% First order

clf
subplot(311)
plot((0:511)./1024,Pxave./max(Pxave))
ylabel('Averaged(abs(fft(x)).^2)')
xlabel('Fraction of sampling frequency')

axes('position',[.45 .78 .15 .12])
plot((130:180)/1024,Pxave(130:180)./max(Pxave(130:180)))
```

```
axis([130/1024 180/1024 0 1])

axes('position',[.7 .78 .15 .12])
plot((285:335)/1024,Pxave(285:335)/max(Pxave(285:335)))
axis([285/1024 335/1024 0 1])

print difffigx

!ps3epsi <difffigx.ps >difffigx.epsi

clf
subplot(311)
plot((0:511)./1024,Pyave./max(Pyave))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

print difffigy
!ps3epsi <difffigy.ps >difffigy.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%
% Second order

Pyave2=sum(PSDy2)./numb;

clf
subplot(311)
plot((0:511)./1024,Pyave2./max(Pyave2))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

axes('position',[.17 .78 .12 .12])
plot((25:75)/1024,Pyave2(25:75)/max(Pyave2(25:75)))
axis([25/1024 75/1024 0 1])

print diff2figy
!ps3epsi <diff2figy.ps >diff2figy.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%
% Third order

Pyave3=sum(PSDy3)./numb;

clf
```

```
subplot(311)
plot((0:511)./1024,Pyave3./max(Pyave3))
ylabel('Averaged(abs(fft(y)).^2)')
xlabel('Fraction of sampling frequency')

axes('position',[.17 .78 .12 .12])
plot((25:75)/1024,Pyave3(25:75)/max(Pyave3(25:75)))
axis([25/1024 75/1024 0 1])

print diff3figy
!ps3epsi <diff3figy.ps >diff3figy.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

APPENDIX Q - MATLAB PROGRAM IMPLEMENTING THE WHITENING PREDICTION ERROR FILTER

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Whitening property of prediction error filter
%
% Author: Capt M.A. Cloutier
% Date:   16 Nov 95
% Name:  adaptfin.m
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clear
figure(1)
clf
orient tall
l=18432;          % length of the data

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Generate the data
%
t=1:l;
s= .475*cos(2*pi*t/8) +0.15*cos(2*pi*t*2/8) + 0.1*cos(2*pi*t*3/8 + pi/3);%
varia=100;          %variance of the AWGN to be colored

w=sqrt(varia)*randn(1,length(s));%white noise

[B,A]=butter(1,0.1);%color the white noise Butter 1st order
n=filter(B,A,w);

x  = s+n;                    % x is the signal + noise

a=input('What filter size is required ?');
mu=0.00005;     %size of the learning parameter

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
w = ones(1,a);    % Initialize filter weights to 1

x = [zeros(1,a-1),x]; %pad input to start data
```

```
nest      = zeros(1,l);%initialize the predicted error vector
skest     = zeros(1,l);%initialize the estimated signal vector

%Estimate the correlation matrix of the noise based on its PSD

[H,P]=freqz(B,A,2047);%H contains the transfer function

Rnt=abs(ifft(abs(H).^2));%with Wiener-Khinchine theorem find Rn

Rs=xcorr(s(1:1024))./1024;%calculate autocorrelation of signal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot both autocorrelation functions to identify maxs and zero crossings

clf
subplot(311)
plot(0:20, Rnt(1:21))
grid
ylabel('Amplitude')
xlabel('Delay')
print adapfigRn
!ps3epsi <adapfigRn.ps >adapfigRn.epsi

clf
subplot(311)
plot((0:20),Rs(1024:1044))
grid
xlabel('Delay')
ylabel('Amplitude')
print adapfigRs
!ps3epsi <adapfigRs.ps >adapfigRs.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Run adaptive LMS algorithm on all data

delay=input('What delay? ');%choose delay based on correlation functions

for i=1:l-delay
   skest(i) = w*flipud(x(i:i+(a-1))');% compute signal est
   nest(i) = x(i+a+delay-1)-skest(i);% compute predicted error
   w = w+mu.*nest(i).*fliplr(x(i:(i+(a-1))));
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Compute averaged periodograms for the input signal, predicted error
% and estimated signal.

PSD1=zeros(10,1024);
PSD2=zeros(10,1024);
PSD3=zeros(10,1024);
index=8193;
for i=1:10
   PSD1(i,:)=abs(fft(x(index:index+1023),1024)).^2;
   PSD2(i,:)=abs(fft(nest(index:index+1023),1024)).^2;
   PSD3(i,:)=abs(fft(skest(index:index+1023),1024)).^2;
   index=index+1024;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output average PSDs

clf
subplot(311)
PSDx=sum(PSD1)./10;
PSDx(1:512)=PSDx(1:512)./max(PSDx(1:512));
plot((0:511)./1024,PSDx(1:512))
ylabel('Averaged(abs(fft(x)).^2)')
xlabel('Fraction of sampling frequency')
axis([0 .5 0 1])

axes('position',[.59 .78 .3 .12])
plot((241:400)/1024,PSDx(241:400)./max(PSDx(241:400)))
axis([241/1024 400/1024 0 1])

print adapfigx
!ps3epsi <adapfigx.ps >adapfigx.epsi

clf
subplot(311)
PSDer=sum(PSD2)./10;
plot((0:511)./1024,PSDer(1:512)./max(PSDer(1:512)))
ylabel('Averaged(abs(fft(x)).^2)')
xlabel('Fraction of sampling frequency')
axis([0 .5 0 1])
print adapfigerr
!ps3epsi <adapfigerr.ps >adapfigerr.epsi
```

```
clf
subplot(311)
PSDsig=sum(PSD3)./10;
plot((0:511)./1024,PSDsig(1:512)./max(PSDsig(1:512)))
ylabel('Averaged(abs(fft(x)).^2)')
xlabel('Fraction of sampling frequency')
axis([0 .5 0 1])

axes('position',[.59 .78 .3 .12])
plot((241:400)/1024,PSDsig(241:400)./max(PSDsig(241:400)))
axis([241/1024 400/1024 0 1])

print adapfigsh
!ps3epsi <adapfigsh.ps >adapfigsh.epsi

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Library, Code 13 2
   Naval Postgraduate School
   Monterey, California 93943-5101

3. Chairman, Code EC 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

4. Prof. Anthony Atchley, Code PH/AY 1
   Chairman Engineering Acoustics Academic Group
   Department of Physics
   Naval Postgraduate School
   Monterey, CA 93943

5. Prof. Ralph Hippenstiel, Code EC/Hi 3
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

6. Prof. Roberto Cristi, Code EC/Cx 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

7. Prof. Monique Farques, Code EC/Fa 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

8. Major Gilbert Larocque, DMAEM 4-3 2
   Building 155, Rockliffe
   National Defence Headquarters
   101 Col By Drive
   Ottawa, (Ontario)
   Canada, K1A 0K2

9. Dennis Guertin, DSIS 3 1
Director Scientific Information Services
National Defence Headquarters
101 Col By Drive
Ottawa, (Ontario)
Canada, K1A 0K2

10. Captain Martin Cloutier, DMAEM 4-3-2 2
Building 155, Rockliffe
National Defence Headquarters
101 Col By Drive
Ottawa, (Ontario)
Canada, K1A 0K2